

# Elim Staiano

## Sommario

Introduzione (Lez.1).....	4
Segnali.....	4
Segnali analogici e digitali.....	5
Immagini digitali.....	5
Discretizzazione.....	5
Tipologie di EID (Elaborazione nelle Immagini Digitali).....	6
Fasi chiave nell'EID.....	7
Tipi di immagini.....	8
Introduzione all'IA e al ML (Lez.2).....	8
Intelligenza Artificiale (definizione e test di Turing).....	9
Sistemi esperti.....	9
Decisioni razionali e agenti.....	10
Machine Learning e apprendimento.....	10
Apprendimento supervisionato.....	11
Apprendimento non supervisionato.....	12
Reinforcement Learning (Apprendimento rinforzato).....	12
Apprendimento supervisionato (Lez.3).....	12
Esempio applicativo (email spam).....	12
Variabili quantitative, qualitative e categoriche ordinate.....	13
Regressione.....	13
Esempio regressione.....	15
Classificazione.....	15
Esempio classificazione.....	17
Apprendimento non supervisionato (Lez.4).....	18
Clustering K-means.....	18
Clustering gerarchico.....	20
Formazione delle immagini (Lez.5).....	22
Struttura dell'occhio umano e formazione delle immagini.....	22
Luminosità percepita.....	23

Immagini digitali e ad alta intensità.....	24
Ottica di base.....	25
Geometria della formazione dell'immagine.....	25
Proiezione prospettica.....	26
Proiezione ortografica.....	26
Lenti sottili.....	27
Fotometria della formazione dell'immagine.....	29
Acquisizione delle immagini (Lez.6).....	30
Radiazioni elettromagnetiche.....	30
Processo di acquisizione delle immagini.....	30
Dispositivi di acquisizione.....	32
Esempio di acquisizione delle immagini.....	33
Modello di formazione dell'immagine.....	33
Campionamento e quantizzazione.....	33
Rappresentazione delle immagini.....	34
Risoluzione spaziale e dell'intensità.....	36
Interpolazione di immagini.....	36
Funzione di mapping.....	37
Relazioni tra pixel.....	37
Vicinanza.....	38
Adiacenza.....	38
Componenti connesse.....	39
Distanza.....	40
Operazioni sulle immagini (Lez.7).....	41
Operazioni puntuali e matriciali.....	41
Operazioni aritmetiche.....	41
Operazioni lineari e non lineari.....	42
Operazioni insiemistiche e logiche.....	42
Operazioni spaziali.....	43
Operazioni su singoli pixel.....	44
Operazioni di vicinanza.....	44
Trasformazioni spaziali geometriche.....	44
Trasformazioni di intensità (Lez.8).....	46

Negativa di un'immagine.....	47
Trasformazioni logaritmiche.....	48
Trasformazioni gamma.....	48
Trasformazioni di stretching del contrasto.....	49
Trasformazioni di selezione (slicing) del livello di intensità.....	50
Trasformazioni bit-plane.....	50
Trasformazioni basate su istogrammi.....	51
Equalizzazione e specifica dell'istogramma.....	52
Elaborazione locale dell'istogramma.....	55
Statistiche da istogramma.....	56
Trasformata di Fourier (Lez.9).....	57
Funzioni trasformate.....	57
Fourier.....	57
Numeri complessi.....	58
Serie di Fourier.....	58
Impulso.....	59
Impulso discreto.....	59
Treno di impulsi.....	60
Trasformata di Fourier continua.....	60
Esempio di trasformata della funzione box.....	61
Esempio di trasformata di un impulso e di un treno di impulsi.....	63
Convoluzione e trasformata di Fourier.....	64
Campionamento e trasformata di Fourier.....	64
Teorema di campionamento.....	66
Aliasing.....	67
Ricostruzione di una funzione campionata.....	68
Trasformata di Fourier discreta (Lez.10).....	69
Convoluzione circolare.....	70
Relazione tra intervalli di campionamento e di frequenza.....	70
Trasformata di Fourier per immagini (2D).....	71
Impulso 2D.....	71
Coppia di trasformate di Fourier continue 2D.....	72
Esempio di trasformata di della funzione box 2D.....	72

Teorema di campionamento 2D.....	72
Aliasing nelle immagini digitali.....	73
Effetto Moirè.....	74
Coppia di trasformate di Fourier discrete 2D.....	74
Proprietà della trasformata di Fourier discreta 2D.....	75
Spettro di Fourier e angolo di fase.....	76
Teorema di convoluzione 2D.....	78
Convoluzione e errore di wraparound.....	79
Filtraggio nel dominio delle frequenze (Lez.11-12).....	79
Filtri passa basso e passa alto.....	80
Wraparound e padding.....	81
Ringing.....	82
Filtri di shift zero phase.....	83
Procedura per il filtraggio delle frequenze.....	83
Progettazione di filtri spaziali.....	84
Filtri passa basso.....	85
Filtri passa alto.....	87
Laplaciano.....	89
Unsharp masking.....	89
Filtraggio omomorfico.....	90
Filtraggio selettivo.....	92
Filtri band-reject e passa-banda.....	92
Filtri notch.....	93
Computazione della trasformata di Fourier discreta.....	94
Filtraggio nel dominio spaziale vs nel dominio della frequenza.....	94

## Introduzione (Lez.1)

L'informatica, in generale, riguarda la rappresentazione, l'elaborazione e la trasmissione delle informazioni, mentre l'elaborazione delle immagini considera, in particolare, l'informazione contenuta nelle immagini digitali.

## Segnali

Nel mondo fisico, una **qualsiasi quantità che esibisca una variazione nel tempo e/o nello spazio è potenzialmente un segnale**. Questi (i segnali) forniscono informazioni sul comportamento o sulla natura di un fenomeno di interesse e possono essere **unidimensionali**, se ad es. rappresentassero la tensione elettrica in un circuito nel tempo, o **bidimensionali**, se ad es. rappresentassero **un'immagine digitale**, dove i due assi corrispondono alle coordinate spaziali.

## Segnali analogici e digitali

In generale un segnale (analogico) viene catturato da un dispositivo di acquisizione (microfono, fotocamera, ecc.), il quale lo converte in un segnale elettrico. Quest'ultimo deve essere poi **discretizzato** attraverso un convertitore analogico/digitale in modo da poterlo successivamente elaborare tramite calcolatore.

Per effettuare invece il passaggio da digitale ad analogico, bisognerà utilizzare un convertitore opposto, ossia digitale/analogico.

## Immagini digitali

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

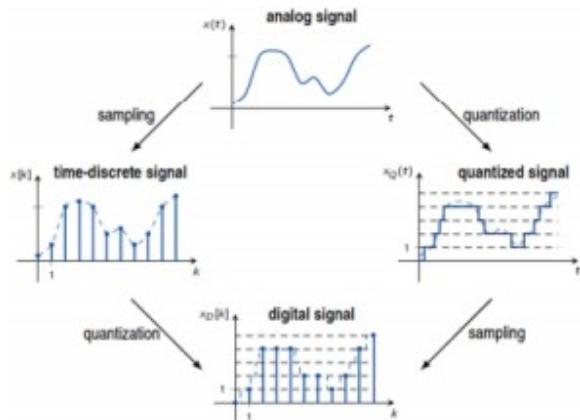
L'immagine è dunque una matrice, composta da un numero finito di elementi, chiamati **pixel**, ciascuno dei quali ha una specifica posizione e un valore corrispondente.

## Discretizzazione

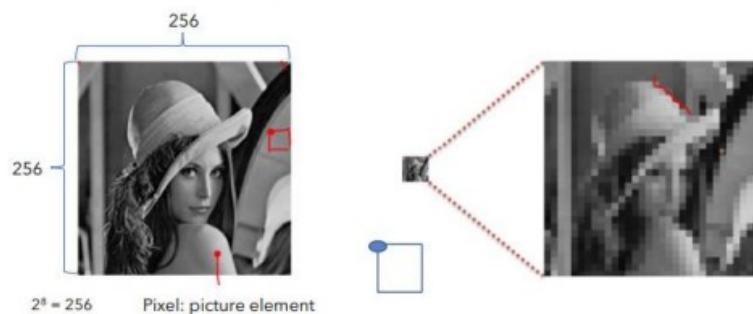
Un'immagine, in realtà, non è altro che un segnale analogico, mentre **un'immagine digitale è un segnale discreto**.

La conversione da segnale analogico a discreto prende il nome di **discretizzazione** e si realizza per mezzo di un'operazione di campionamento ed una di quantizzazione:

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



CAMPIONAMENTO (SAMPLING)



QUANTIZZAZIONE



Il livello di dettaglio dipende, come si può vedere dalle immagini sopra, dal numero di livelli di grigio rappresentabili (256, 16, 4 negli esempi sopra).

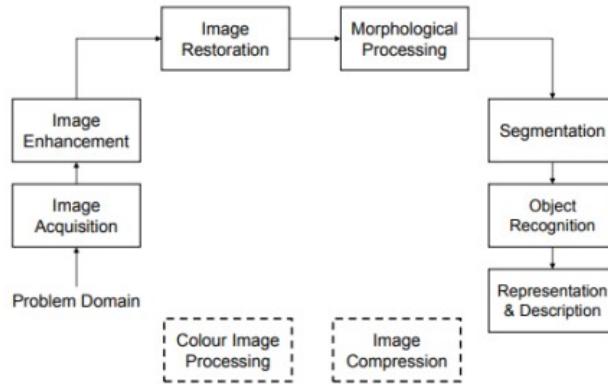
## Tipologie di EID (Elaborazione nelle Immagini Digitali)

Nell'EID ci sono almeno **tre discipline** che si sovrappongono: elaborazione delle immagini, comprensione delle immagini e computer vision. Il confine tra queste tre discipline non è ben definito, ma possiamo dire che:

- **l'elaborazione delle immagini** ha come fine il miglioramento delle immagini, di conseguenza sia come input che come output avrà delle immagini;
- **la comprensione delle immagini** si occupa dell'estrazione di caratteristiche o feature, quindi data un'immagine in input, si otterranno in output degli attributi;
- **la computer vision**, che è un settore dell'IA, che esegue operazioni di riconoscimento di oggetti. In pratica prende in input degli attributi, fornendo in output una "comprensione" degli stessi.

In generale l'EID si focalizza sul miglioramento dell'informazione grafica per l'interpretazione umana e sull'elaborazione dei dati dell'immagine per scopi di memorizzazione, trasmissione e rappresentazione per la percezione autonoma delle macchine.

## Fasi chiave nell'EID



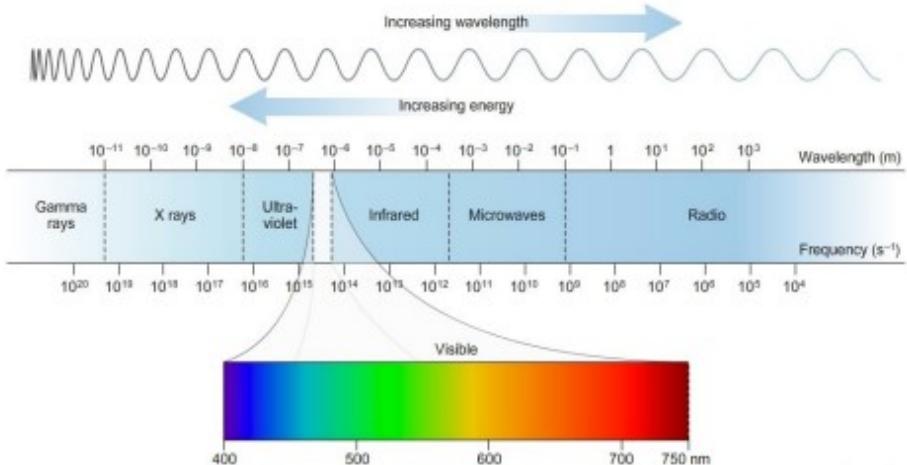
Esistono diverse fasi tipiche dell'elaborazione delle immagini digitali:

- **acquisizione dell'immagine**, che riguarda la generazione dell'immagine a partire da una sorgente, come può essere la luce;
- **miglioramento dell'immagine**, che è un processo di manipolazione di un'immagine al fine di migliorarne la qualità (è soggettivo);
- **ripristino dell'immagine**, che si basa su tecniche di ripristino basate su modelli matematici (è oggettivo a differenza del miglioramento dell'immagine);
- **elaborazione morfologica**, che permette di estrarre le componenti di un'immagine;
- **segmentazione**, che divide l'immagine nelle sue parti costituenti.

Dopo la segmentazione potrebbero anche essere eseguite operazioni di **riconoscimento degli oggetti**, con conseguente **rappresentazione e descrizione**, ma in questo caso si parlerebbe di tecniche legate all'IA.

Abbiamo poi anche altre fasi importanti, ma slegate dal ciclo vero e proprio, come **l'elaborazione di un'immagine a colori** e la **compressione**.

## Tipi di immagini



- Basate su radiazioni dallo spettro Elettromagnetico (EM)

- Acustiche/ultrasoniche
- Elettroniche
- Sintetiche

Esistono diversi tipi di immagini che derivano dall'utilizzo di dispositivi che riescono ad intercettare segnali elettromagnetici, a differenza dell'occhio umano che è in grado di percepire solo una piccola banda di frequenza, ossia la luce visibile (regione bianca nell'immagine).

In base alla loro natura le immagini si classificano in:

- **immagini di riflessione**, che contengono informazioni sulla superficie degli oggetti (imaging ottico, onde radio, laser);
- **immagini di emissione**, che contengono informazioni interne all'oggetto (immagini termiche o infrarosse);
- **immagini di assorbimento**, che contengono informazioni sulla struttura interna all'oggetto (immagini a raggi X).

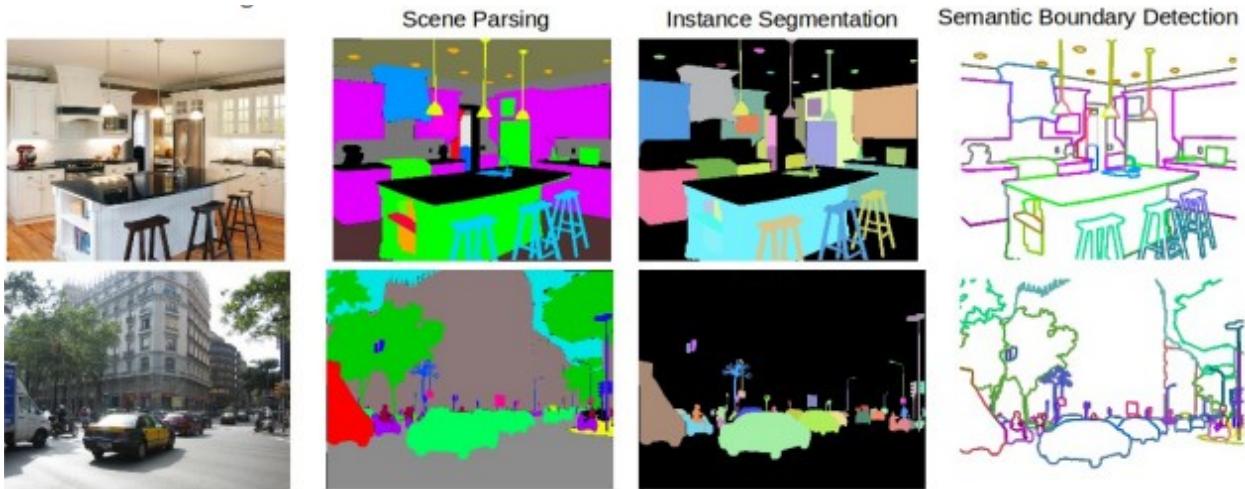
Sono molti e diversi tra loro i campi in cui le informazioni vengono acquisite sotto forma di immagini. Alcuni campi applicativi, abbastanza classici, sono la medicina, l'industria, l'astronomia, ecc.

## Introduzione all'IA e al ML (Lez.2)

L'Intelligenza Artificiale e il Machine Learning si occupano di operazioni di riconoscimento di oggetti e volti, segmentazione di scene e classificazione di immagini.

Una normale **segmentazione** distingue solo i contorni degli oggetti, mentre una **segmentazione semantica** permette anche di riconoscere il significato degli stessi (oggetti). La segmentazione semantica comprende:

- **scene parsing**, dove tutte le istanze di una stessa classe di oggetti avranno lo stesso colore;
- **instance segmentation**, dove anche istanze diverse della stessa classe avranno colori diversi;
- **semantic boundary detection**, dove viene effettuata una segmentazione tenendo conto solo dei bordi e non della parte interna.



## Intelligenza Artificiale (definizione e test di Turing)

Nel corso del tempo sono state date diverse definizioni al concetto di **Intelligenza Artificiale**, che si possono racchiudere in:

- sistemi che pensano come esseri umani;
- sistemi che agiscono come esseri umani;
- sistemi che pensano in modo razionale;
- sistemi che agiscono in modo razionale.

Per verificare se un sistema si comporta in modo intelligente potrebbe essere utilizzato il **test di Turing**. Questo consiste in un test che coinvolge tre partecipanti, ossia un interrogatore, umano, una macchina e un altro essere umano.

L'interrogatore pone domande sia alla macchina che all'essere umano e se la prima è in grado di fargli credere che stia interagendo con l'essere umano allora supererebbe il test. In realtà tale test non può essere considerato come attendibile per giudicare l'intelligenza di un sistema in quanto non considera molti aspetti percepiti come intelligenza, come ad es. la percezione, la strategia, ecc.

## Sistemi esperti

I sistemi esperti sono programmi progettati per emulare la capacità di un esperto umano di risolvere problemi specifici in un determinato dominio di conoscenza.

I sistemi esperti sono composti da:

- **base di conoscenza**, che contiene informazioni essenziali sul dominio del problema ed è rappresentata con fatti e regole codificate mediante costrutti IF – THEN;
- **motore inferenziale**, che è un meccanismo utilizzato per derivare nuova conoscenza a partire dalla base di conoscenza e dalle informazioni fornite dall'utente;

- **interfaccia utente**, che permette l'interazione con gli utenti, oltre allo sviluppo e alla gestione della base di conoscenza.

## Decisioni razionali e agenti

Il termine **razionale** viene utilizzato per indicare il raggiungimento degli obiettivi prefissati nel miglior modo possibile, di conseguenza significa massimizzare la propria utilità attesa.

Il concetto di razionalità va spesso di pari passo con quello dell'**agente**, che in generale è un'entità che percepisce e agisce. Gli **agenti razionali**, invece, sono un particolare tipo di agenti che seleziona le azioni che massimizzano la sua utilità attesa.

La razionalità delle azioni viene influenzata da caratteristiche quali la percezione, l'ambiente e lo spazio delle stesse (azioni).

## Machine Learning e apprendimento

Il Machine Learning (ML) è un **sottoinsieme dell'IA che ha lo scopo di dotare i computer della capacità dell'uomo di apprendere dall'esperienza**.

Per raggiungere questo obiettivo vengono utilizzate tecniche che permettono di apprendere le informazioni dai dati, con più esempi (dati) che equivalgono ad una migliore prestazione da parte dei computer.

Il tema centrale delle applicazioni di ML è l'**apprendimento**, che può essere **supervisionato, non supervisionato o rinforzato**.

In generale, per definire l'apprendimento, possiamo dire che un programma apprende dall'**esperienza**, rispetto ad una determinata **classe di compiti** e ad una **misura delle prestazioni**, se le sue prestazioni sui compiti migliorano con l'esperienza.

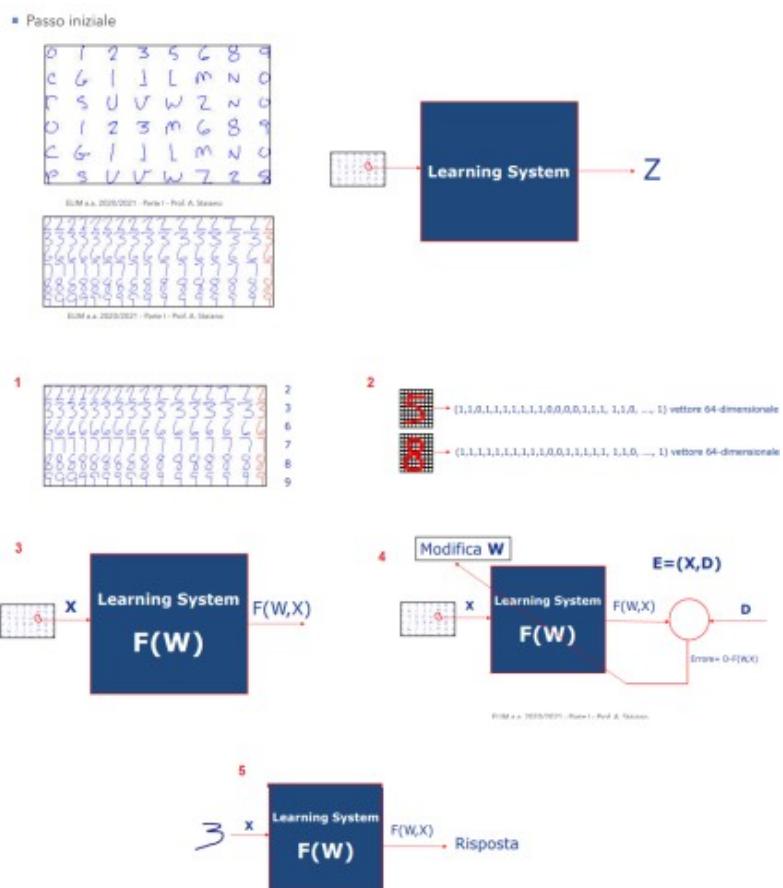
Per progettare un sistema di apprendimento bisogna seguire alcuni passi, ma come prima cosa consideriamo lo stesso (sistema di apprendimento) come se fosse una **black-box**, che in base all'**input** dato restituisce un valore predetto. I passi da seguire sono:

1. **raccogliere esempi per il training set**, che riguardano l'esperienza, in quanto senza dati il sistema non sarebbe in grado di apprendere;
2. **rappresentare l'esperienza**, cioè si deve scegliere uno schema di rappresentazione per gli esempi (come può essere un vettore ad es.);
3. **scegliere una rappresentazione per la black-box**, che consiste nello scegliere una funzione che in base all'input ricevuto dalla black-box restituisca una classificazione dello stesso (input). Questa funzione sarà,

ovviamente, influenzata da parametri adattivi, o pesi, che l'algoritmo di apprendimento può imparare;

4. **apprendere/modificare i pesi**, ossia è necessario un algoritmo di apprendimento che permetta appunto di modificare i pesi come conseguenza dell'esperienza maturata a partire dal training set;
5. **testare il sistema**, quindi a partire da un nuovo input, il sistema calcola la sua risposta (previsione) in base alla funzione scelta nel punto 3.

Es. di sistema di apprendimento relativo al riconoscimento di caratteri:



## Apprendimento supervisionato

Un algoritmo di apprendimento supervisionato prende un **insieme di dati di input noti**, anche detti **feature**, ed un **insieme di risposte agli stessi** (sempre note ovviamente), addestrando un modello per generare previsioni accurate per la risposta a successivi nuovi dati.

I modelli predittivi possono essere sviluppati mediante:

- **tecniche di classificazione**, che predicono risposte discrete, ad es. se un email è spam o meno, classificando i dati di input in categorie o classi;
- **tecniche di regressione**, che predicono risposte continue, come ad es. i cambiamenti della temperatura.

## Apprendimento non supervisionato

L'apprendimento non supervisionato si occupa di trovare **pattern nascosti o strutture intrinseche nei dati**, derivando inferenze dai dataset (insieme dei dati di input) che consistono solo di dati di input senza risposte etichettate.

La più comune forma di apprendimento non supervisionato è il **clustering**, che appunto raggruppa i dati per caratteristiche comuni.

## Reinforcement Learning (Apprendimento rinforzato)

Il Reinforcement Learning (RL) è un paradigma di apprendimento finalizzato alla costruzione di **agenti software che intraprendono azioni in un ambiente al fine di massimizzare un profitto espresso in forma di premio**. In pratica se l'agente compie azioni corrette allora gli viene assegnato un "premio", altrimenti gli verrà data una "punizione".

## Apprendimento supervisionato (Lez.3)

Introduzione all'apprendimento supervisionato.

Nell'apprendimento supervisionato il punto di partenza è rappresentato da un insieme di dati disponibili, chiamato **training set**, in base ai quali costruire il modello predittivo.

## Esempio applicativo (email spam)

- Costruire un sistema di individuazione di email spam
  - Dati da 4061 email, ognuna è etichettata come **spam** o **email**
  - Obiettivo: costruire un filtro di spam personalizzato
  - Input (caratteristiche): frequenze relative di 57 parole che occorrono più comunemente e simboli di interruzione in tali messaggi di email

	george	you	hp	free	!	edu	remove
spam	0.00	2.26	0.02	0.52	0.51	0.01	0.28
email	1.27	1.27	0.90	0.07	0.11	0.29	0.01

In questo esempio abbiamo nella tabella la frequenza di alcune delle 57 parole che occorrono più comunemente in tutte le email. In particolare viene specificata la loro frequenza, sul totale delle parole presenti nelle email, sia per quelle spam che per quelle normali.

## Variabili quantitative, qualitative e categoriche ordinate

Le variabili, sia di input che di output, possono essere **quantitative**, nel caso in cui assumano un valore numerico misurabile (ad es. riguardano le altezze degli individui: 160 cm, 170 cm, 180 cm, ecc.), oppure **qualitative**, nel caso in cui invece rappresentino categorie o qualità distinctive senza un ordine intrinseco (ad es. le categorie potrebbero essere: blu, verde, marrone, ecc. ), quindi non quantificabili su scala numerica.

**La distinzione nel tipo di output determina lo specifico compito di predizione:**

- **regressione**, nel caso in cui si vogliono predire output quantitativi;
- **classificazione**, nel caso in cui si vogliono predire output qualitativi.

La **classificazione delle variabili per quanto riguarda gli input**, invece, porta ad una **distinzione dei metodi da poter utilizzare per la predizione**, in quanto alcuni potrebbero essere più adatti per gli input quantitativi o viceversa.

Esiste anche un terzo tipo di variabile, anche in questo caso sia per gli input che per gli output, detta **categorica ordinata**. In questo caso vi è un ordinamento tra i valori, ad es. le variabili possono assumere valori “piccolo”, “medio” o “grande”, ma non c’è una metrica adeguata tra loro.

Le **variabili qualitative**, sia di input che di output, vengono rappresentate mediante **codici numerici**. Il caso più semplice si ha con due categorie, o classi, (ad es. “successo” o “fallimento”), rappresentate solitamente con valori 0 e 1, che prendono il nome di target.

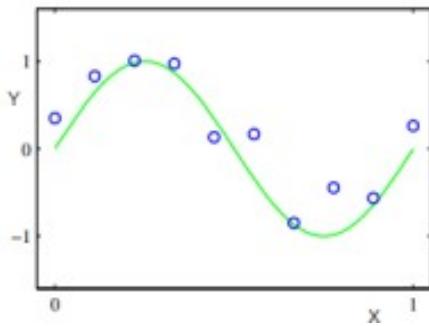
## Regressione

L’obiettivo della regressione è quello di **sfruttare il training set**, composto dalle coppie formate dai dati di input associati all’output previsto, **per predire i valori delle variabili relative agli output quantitativi Y, corrispondenti a nuovi valori di input X.**

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

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

con  $f$  una funzione fissata ma non nota ed  $\epsilon$  che rappresenta un errore casuale additivo.



Per effettuare la predizione possiamo considerare un semplice approccio basato sul **fitting** (approssimazione di un modello matematico ai dati osservati, in modo da trovare la miglior rappresentazione per poi effettuare previsioni sui dati futuri) di curve, determinandone una che si adatta alla rappresentazione del training set. Supponiamo di utilizzare una **funzione polinomiale della forma**

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

con **w** il **vettore relativo ai coefficienti del polinomio**, che verranno determinati dalla procedura di fitting. Si procede quindi alla **minimizzazione di una funzione di errore E** che misura la discrepanza del fitting tra la funzione polinomiale appena vista  $y(x, w)$ , per una qualsiasi combinazione dei valori del vettore relativo ai coefficienti del polinomio **w**, ed i punti nel training set. Una semplice funzione di errore è la **somma dei quadrati degli errori tra le predizioni per ogni dato ed il corrispondente valore di output previsto**

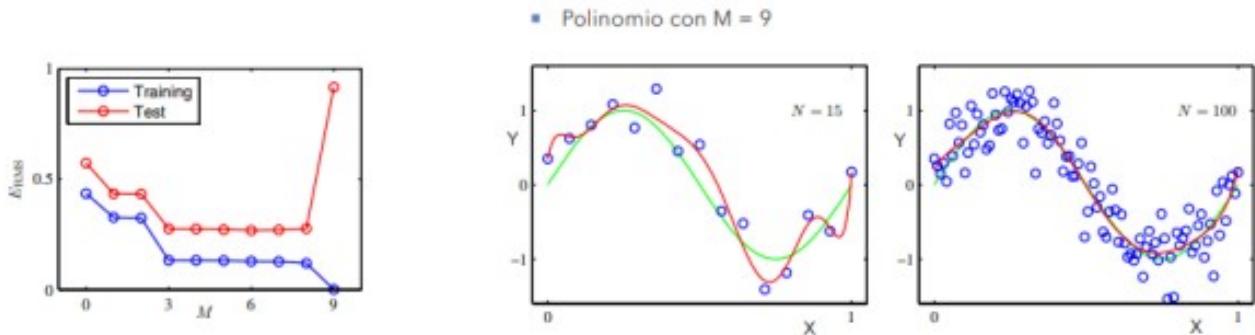
$$E(w) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, w) - y_n\}^2$$

Ricapitolando, quindi, si può risolvere il problema del fitting della curva scegliendo la combinazione dei valori del vettore dei coefficienti del polinomio **w** per cui la funzione di errore sia la più piccola possibile. Quest'ultima è una funzione quadratica rispetto ai coefficienti **w**, per cui la sua derivata rispetto agli stessi (coefficienti) è lineare e, di conseguenza, la **minimizzazione della funzione di errore ha una soluzione unica**.

Di fondamentale importanza risulta essere la **scelta del grado del polinomio da utilizzare nel fitting**, infatti in linea generale **maggior è il grado del polinomio e più accurata sarà la predizione**, tuttavia si rischia di creare **overfitting**. Ad es. se il numero di dati nel training set è uguale al grado del polinomio significa che la funzione di predizione passerà per tutti i punti, ma **noi cerchiamo una generalizzazione** che ci permetta di eseguire predizioni accurate per nuovi dati e non un'interpolazione.

Per valutare oggettivamente se il modello effettua una buona **generalizzazione**, si utilizza un nuovo insieme di dati, detto **test set**, che **conterrà solo variabili di input**, in modo da valutare la predizione. In pratica si procede a tentativi, valutando l'errore che consegue dai diversi valori del grado del polinomio.

In generale per diminuire l'overfitting la soluzione migliore consiste nell'aumentare i dati presenti nel training set.



## Esempio regressione

### ESEMPIO REGRESSIONE

- Dati relativi a diversi modelli di auto, dal 1970 al 1982
    - 406 auto
    - 7 feature (variabili di input),  $\mathbf{X} = (x_1, x_2, \dots, x_7)^T$ 
      - $X = (\text{Acceleration}, \text{Cylinders}, \text{Displacement}, \text{Horsepower}, \text{Model\_year}, \text{Weight}, \text{Origin})$
      - 1 target (variabile di output)  $Y = \text{MPG}$
    - Predire il consumo di carburante (espresso in miglia per galloni)
    - Statisticamente, anche in questo caso vogliamo apprendere il seguente modello
- $$Y = f(\mathbf{X}) + \epsilon$$

Trattandosi di un esempio della tecnica di regressione, la variabile di output è di tipo quantitativo, quindi misurabile numericamente.

## Classificazione

L'obiettivo della classificazione è **prendere un campione di input e assegnarlo ad una di K classi discrete**. Geometricamente è come **dividere lo spazio di input in regioni di decisione** i cui confini sono chiamati **confini di decisione**.

Nel caso in cui il **numero di classi discrete sia uguale a 2** allora si parla di **classificazione binaria**, altrimenti di **classificazione multi-classe**.

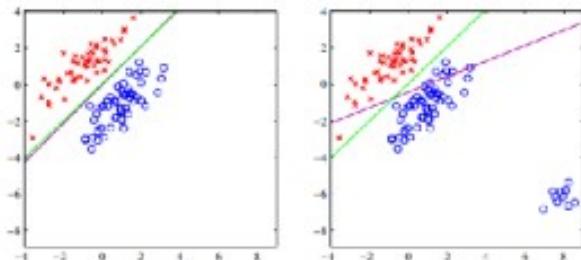
Nei problemi di classificazione ci sono diversi modi per utilizzare i valori **target** in modo da rappresentare una classe. Ad es. **in caso di classificazione binaria** (ossia due classi) in cui c'è un unico target, questo potrebbe assumere **l'identificatore della classe a cui viene assegnato il campione di input**. Un'altra possibilità consiste nell'interpretare il valore **target come la probabilità di una determinata classe**, che in ambito binario potrebbe assumere solo i valori 0 e 1.

Quando si parla di **classificazione multi-classe** è possibile adottare lo **schema di codifica 1-a-K** in cui il target sarà un vettore di lunghezza K che avrà valore 1

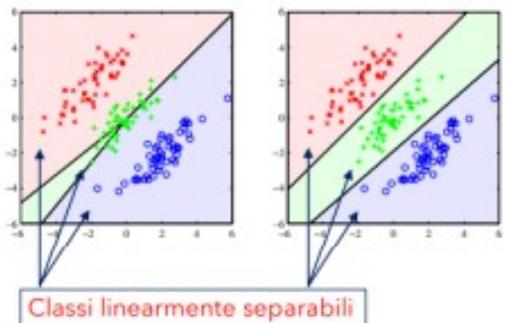
**solo nella componente relativa alla classe da predire**, con valori 0 in tutte le altre posizioni.

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

CLASSIFICAZIONE BINARIA: CONFINI DI DECISIONE

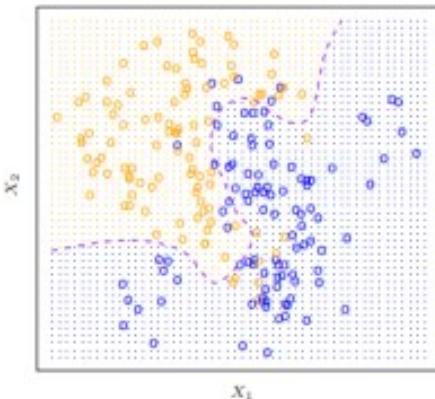


CLASSIFICAZIONE MULTI-CLASSE



CLASSIFICAZIONE

\* Classi **non** linearmente separabili

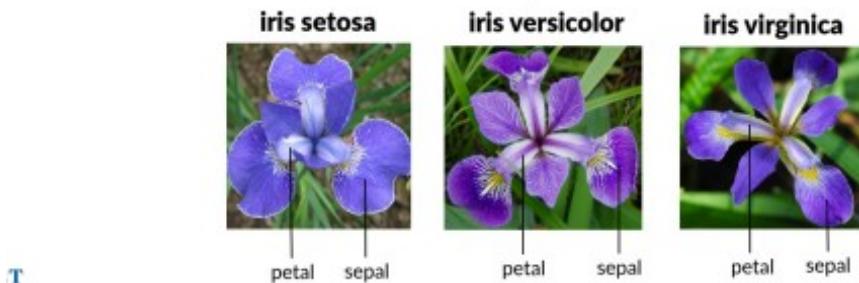


Esistono **diversi approcci alla classificazione**, tra cui uno potente e molto utilizzato che consiste nel modellare la distribuzione di probabilità in fase di inferenza, prendendo decisioni di classificazione in base ad essa (la distribuzione).

## Esempio classificazione

### ESEMPIO CLASSIFICAZIONE

- Misure caratteristiche di tre specie di fiore Iris
  - 150 fiori
  - 4 feature:  $\mathbf{X} = (x_1, x_2, x_3, x_4)^T$ 
    - $\mathbf{X} = (\text{L\_Petal}, \text{A\_Petal}, \text{L\_Sepalo}, \text{A\_Sepalo})$
  - 3 output: Specie di Iris,  $C_1 = \text{Versicolor}$ ,  $C_2 = \text{Virginica}$ ,  $C_3 = \text{Setosa}$
- L'obiettivo è classificare correttamente le tre specie di Iris



Nell'ambito delle tecniche di classificazione può essere utilizzata la **matrice di confusione** per confrontare le prestazioni di un modello con i valori target. In pratica sulle righe vengono rappresentate le classi prese dal dataset, mentre sulle colonne le predizioni del modello. Le risposte corrette saranno quelle rappresentate sulla diagonale principale e per sapere la percentuale di correttezza si devono sommare i risultati corretti (quelli sulla diagonale appunto), dividendoli poi per il numero totale di valori in input.

Es. di matrice di confusione per l'esempio di Iris.

		Predicted class		
		setosa	versicolor	virginica
True class	setosa	50	0	0
	versicolor	0	48	2
	virginica	0	0	49

## Apprendimento non supervisionato (Lez.4)

### Introduzione all'apprendimento non supervisionato.

Nell'apprendimento non supervisionato, a differenza di quello supervisionato, **non vengono fornite anche le soluzioni previste all'interno del training set**, che quindi **consta solo di dati di input**. Ciò rende l'apprendimento non supervisionato più difficile di quello supervisionato, non avendo un riferimento preciso (variabili di output).

**Spesso è più semplice ottenere dati non etichettati** dalla strumentazione di un laboratorio o da un computer rispetto a dati etichettati che richiedono l'intervento umano. Ad es. le tecniche non supervisionate potrebbero essere utilizzate per formare gruppi di acquirenti in base alle loro storie di acquisto e visualizzazione di prodotti.

Questo tipo di apprendimento **viene anche utilizzato per operazioni di pre-processing** prima di applicare tecniche di apprendimento supervisionato.

Come detto in precedenza, il **clustering** è una tecnica, o un insieme di tecniche, applicata per **individuare sottogruppi**, anche detti **cluster**, **in un data set**. I gruppi saranno formati in modo tale che le **osservazioni siano abbastanza simili tra loro**, tuttavia ciò significa che bisogna definire il concetto di **similarità e diversità**.

Esistono diversi tipi di clustering, ma i due più utilizzati risultano essere il **K-means** ed il **clustering gerarchico**. In generale è bene tenere a mente che **non esiste una sola risposta corretta** per quanto riguarda le scelte da prendere nell'utilizzo delle varie tecniche di clustering (numero di cluster, dissimilarità, linkage, ecc.).

Si provano quindi diverse soluzioni, prendendo infine quella più utile o interpretabile.

### Clustering K-means

Nella tecnica di clustering K-means **si cerca di partizionare le osservazioni (input) in un numero predefinito di cluster K**.

Questi insiemi (cluster) devono soddisfare **due proprietà**:

- se faccio l'**unione** di tutti gli insiemi devo ottenere tutte le osservazioni, e ciò significa che **ogni osservazione appartiene almeno ad uno dei K cluster**;
- se faccio l'**intersezione** di due cluster, per qualsiasi coppia di cluster, devo ottenere un'intersezione vuota. Ciò significa che non ci sarà **nessuna osservazione appartenente a più cluster**.

L'obiettivo che l'algoritmo si propone è quello di **minimizzare la variabilità all'interno di ciascun cluster**, che sarebbe ovviamente la misura di quanto le osservazioni in un cluster differiscono tra loro. Di solito per definire la variabilità intra-cluster viene utilizzata la **distanza euclidea**, che rappresenta la lunghezza del

percorso più breve tra due punti nello spazio. Combinando la minimizzazione della variabilità intra-cluster al calcolo di quest'ultima mediante la distanza euclidea si ottiene la cosiddetta **funzione obiettivo** dell'algoritmo di clustering K-means:

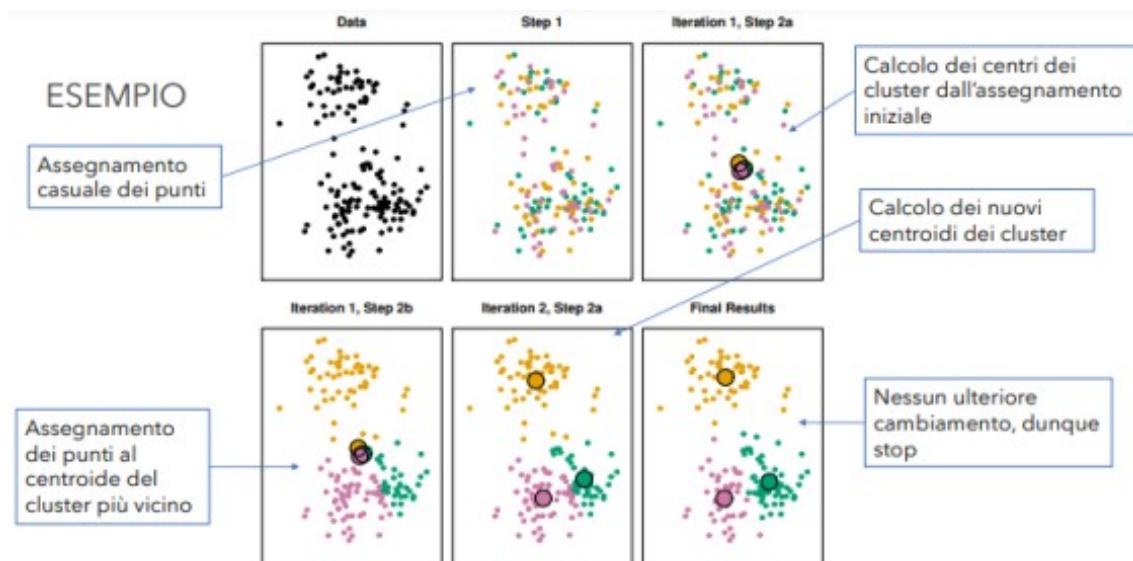
$$\min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

**Funzione obiettivo**

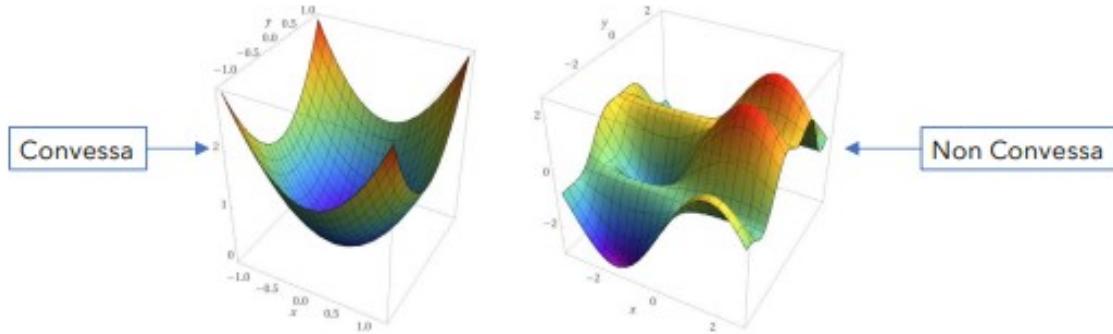
Ricapitolando, i **passi da seguire** nell'algoritmo sono:

1. **assegnare casualmente un numero**, da 1 a K, ad ogni osservazione, che serve come assegnamento iniziale delle osservazioni ai cluster;
2. **iterare fino a che l'assegnamento ai cluster rimane stabile**, ossia non cambia più, e per ognuno dei K cluster **calcolare il centroide**, che rappresenta il punto centrale del gruppo di punti, quindi di osservazioni, assegnati a quel cluster;
3. **assegnare ora ogni osservazione al cluster il cui centroide è più vicino**, con la vicinanza che viene misurata mediante la distanza euclidea.

Es. clustering K-means:



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



Per questo motivo il K-means può rimanere bloccato in **ottimi locali**, non trovando la migliore soluzione. Di conseguenza è **importante eseguire l'algoritmo più volte**, iniziando con un assegnamento casuale diverso ogni volta.

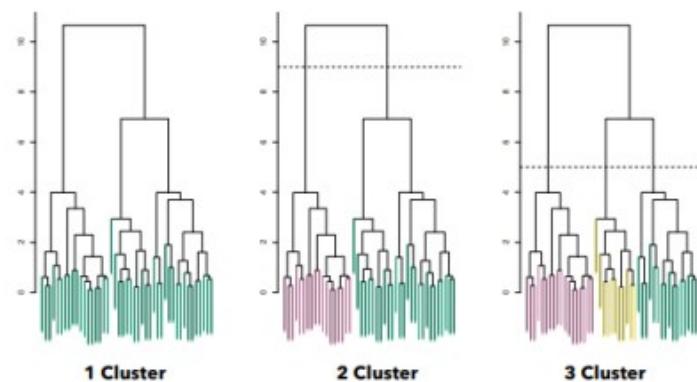
## Clustering gerarchico

Nel clustering gerarchico **non viene definito un numero di cluster e si ottiene 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.

Il tipo più comune di clustering gerarchico è il **bottom-up**, o **agglomerativo**, con questo termine che si riferisce al fatto che il dendrogramma viene costruito a partire dalle foglie, combinando i cluster fino alla radice.

L'algoritmo comincia dall'**inserimento di ogni osservazione in un cluster differente** e si procede al raggruppamento, a due a due, dei **cluster vicini**. Dopodichè si ripete questo procedimento fino a quando tutti i punti si troveranno in un unico cluster.

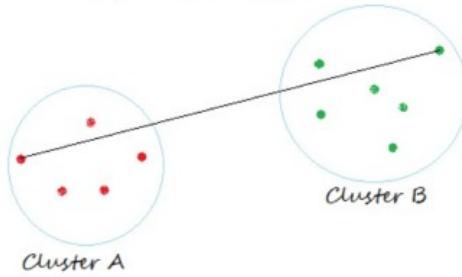
Per scegliere il numero di cluster si traccia una linea che attraversa il **dendrogramma**, in modo da sceglierne un qualsiasi numero in base al punto in cui effettuiamo un "taglio".



L'operazione chiave di questa tecnica di clustering è però il **calcolo della dissimilarità**, o collegamento (**linkage**), che permette di fondere dei cluster invece di altri. Esistono **4 opzioni diverse**, basate sulla **distanza euclidea**:

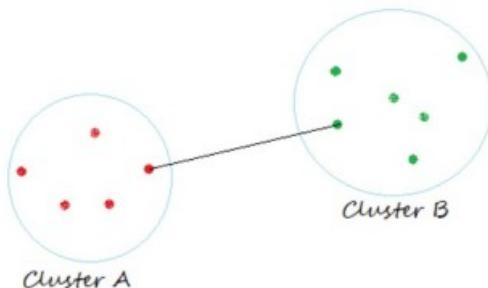
- **complete linkage**, nel quale viene scelta la coppia di cluster che ha la minima distanza più grande tra le osservazioni;

*Complete Linkage*



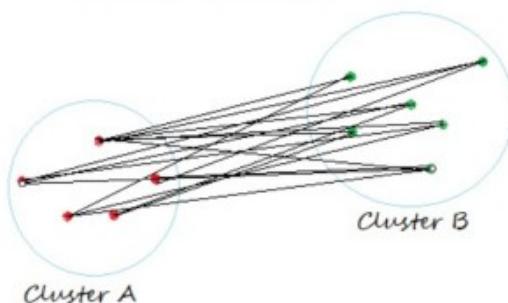
- **single linkage**, nel quale viene scelta la coppia di cluster che ha la minima distanza più piccola tra le osservazioni;

*Single Linkage*



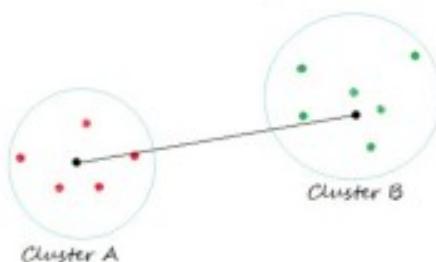
- **average linkage**, nel quale viene scelta la coppia di cluster che ha la minima distanza media tra le osservazioni;

*Average Linkage*

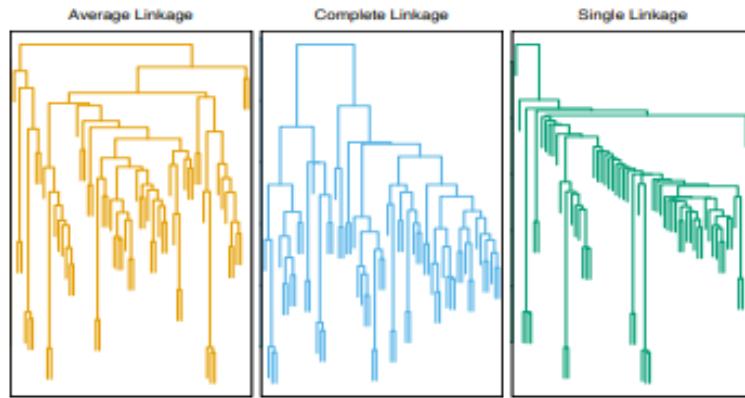


- **centroide**, nel quale viene scelta la coppia di cluster che ha la minima distanza tra i centroidi;

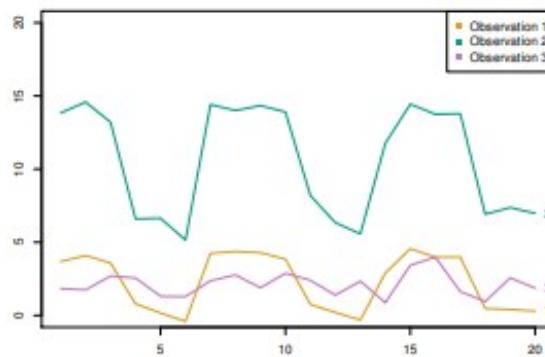
*Centroid Linkage*



In base al metodo di linkage scelto si possono avere risultati molto diversi, infatti ad es. il complete e l'average tendono a costruire cluster di dimensioni uguali, mentre il single porta a cluster estesi.



Per quanto riguarda la misura della dissimilarità abbiamo visto casi relativi alla distanza euclidea, che però non sempre restituisce il miglior risultato. Un'alternativa è la **distanza basata su correlazione**, che considera due osservazioni simili se i loro dati di input sono altamente correlati. Nell'esempio sottostante possiamo vedere che in termini di distanza euclidea le osservazioni 1 e 3 sono simili, ma le osservazioni 1 e 2 risultano altamente correlate, quindi simili in termini di distanza basata sulla correlazione.



## Formazione delle immagini (Lez.5)

### Struttura dell'occhio umano e formazione delle immagini

L'occhio umano ha una forma pressochè sferica, con un diametro complessivo di circa 20mm. Questo (l'occhio umano) è circondato da **tre membrane**:

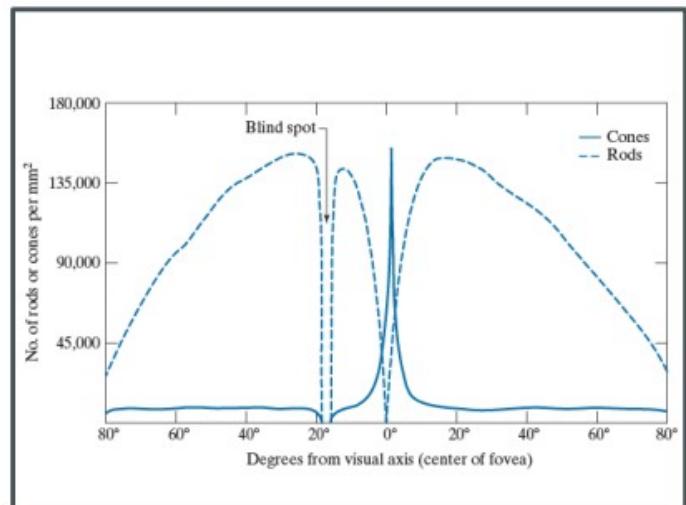
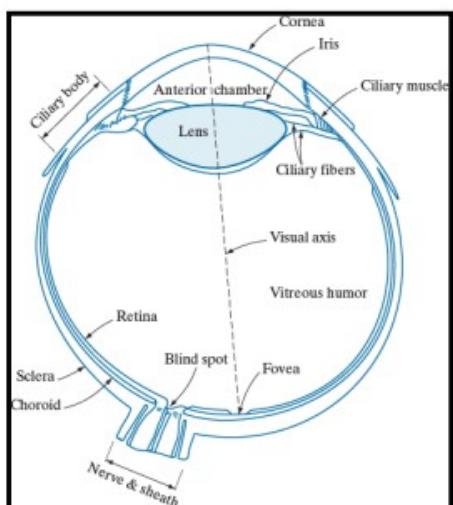
- la **cornea** e la **sclera**, che sono le due membrane più esterne, con la seconda che è una continuazione della prima. La cornea è la parte trasparente all'esterno dell'occhio che copre l'**iride** e la **pupilla**, mentre la sclera è una membrana opaca;
- la **coroide**, che si trova sotto la sclera ed è di colore scuro. Ha lo scopo di limitare la quantità di luce esterna che entra nell'occhio;
- la **retina**, che è la membrana più interna dell'occhio. Questa (la retina) permette la visione mediante recettori di luce distribuiti sulla propria superficie.

I recettori di luce sulla retina sono:

- i **coni**, che sono **molti sensibili ai colori** e per questo sono legati alla **visione fotopica**, ossia quella **diurna**. Ne abbiamo circa 6-7 milioni posizionati intorno alla parte centrale della retina, chiamata **fovea**, ed **ognuno di essi è connesso ad una cellula nervosa**;
- i **bastoncelli**, che sono **poco sensibili ai colori** e per questo sono legati alla **visione scotopica**, ossia quella **nottturna**. Ne abbiamo circa tra i 75 ed i 150 milioni distribuiti su tutta la retina, ma più densamente intorno al centro dell'occhio, e si ha una **relazione 1 a N con le cellule nervose**.

Nei **coni**, avendo una corrispondenza 1 a 1 con le cellule nervose, sarà possibile avere una **miglior distinzione (discriminazione) tra gli oggetti** o le caratteristiche di un'immagine rispetto ai bastoncelli. Questo perchè i coni si concentrano intorno alla parte centrale della retina (la fovea) e ciò si traduce in una maggiore nitidezza.

Vicino alla fovea si ha, inoltre, un punto nel quale sono assenti i recettori, detto **punto cieco**.

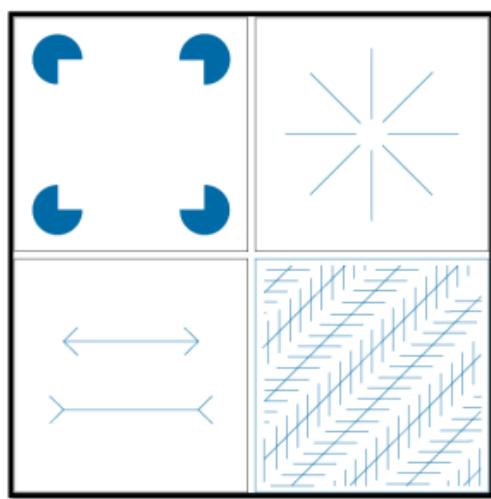


La formazione delle immagini nell'occhio umano avviene grazie al **cristallino**, che è una lente capace, grazie all'azione del **muscolo ciliare**, di modificare la sua forma per **regolare la messa a fuoco** dei raggi luminosi sulla retina, **in base alla distanza dell'oggetto**. Il muscolo ciliare, in particolare, ispessisce il cristallino in corrispondenza di oggetti distanti o lo assottiglia in corrispondenza di oggetti vicini.

## Luminosità percepita

L'occhio umano è in grado di percepire un intervallo molto grande di livelli di intensità di luce, pari a  $10^{10}$  livelli distinti. In ogni istante, però, può essere distinto solo un sottoinsieme di livelli vicini alla luminosità della scena, con il range di sensibilità che varia adattandosi ai cambiamenti di luminosità.

**La luminosità percepita non dipende solo dall'intensità di luce ma anche dalla luminosità ambientale.** Basti pensare al **contrasto** (prima figura sottostante), che fa percepire oggetti dello stesso colore, su sfondi diversi, come oggetti con colori leggermente diversi, o anche **illusioni ottiche** (seconda figura sottostante), ossia percezioni che rappresentano in modo erroneo la realtà osservata.



## Immagini digitali e ad alta intensità

Le immagini digitali vengono ottenute a partire da un dispositivo di imaging, che produce specifiche tipologie di immagini. Una particolare tipologia è rappresentata dalle **immagini ad alta intensità**, le quali **misurano la quantità di luce che incide in un dispositivo fotosensibile**.

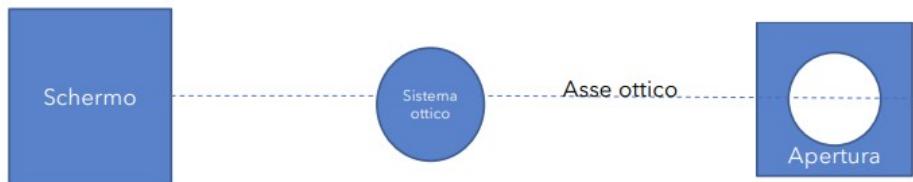
Il processo di formazione di questo tipo di immagini **prevede diversi parametri**:

- **parametri geometrici**, che determinano la posizione dell'immagine su cui è proiettato un punto 3D e includono il tipo di proiezione utilizzata, la posizione e l'orientamento della fotocamera nello spazio e le distorsioni prospettiche introdotte durante il processo di imaging;
- **parametri ottici**, che caratterizzano le ottiche del sensore e includono la lunghezza focale, l'apertura angolare, il campo di visione, ecc.;
- **parametri fotometrici**, che riguardano l'energia luminosa che raggiunge il sensore dopo che questa (l'energia luminosa) è stata riflessa dagli oggetti della scena. In particolare riguardano tipo, intensità e direzione di illuminazione, ma anche proprietà di riflettanza delle superfici osservate.

## 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:

- riducendo l'**apertura della camera ad un punto chiamato pinhole**, con solo un raggio proveniente da un qualsiasi punto della scena che può entrare nella camera;
- introducendo un **sistema ottico composto da lenti**, con i raggi provenienti da uno stesso punto 3D che convergono su un singolo punto dell'immagine.



## Geometria della formazione dell'immagine

Il modello geometrico più semplice della formazione dell'immagine è la **telecamera pinhole** (foro stenopeico).

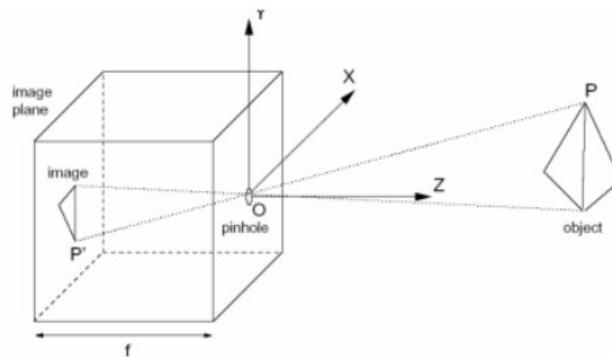
Considerando P un punto della scena di coordinate  $(X, Y, Z)$ ,  $P'$  la sua proiezione sul piano immagine di coordinate  $(X', Y', Z')$  e  $f$  la **distanza focale**, ossia la distanza del centro di proiezione O (pinhole) dal piano immagine, allora dalla **similarità dei triangoli** si ottiene che i punti P, O e  $P'$  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 possiamo modellare la proiezione prospettica ponendo il piano immagine davanti al centro di proiezione, eliminando così il segno negativo.

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

## Proiezione prospettica

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

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:

$$(X_0 + \lambda U, Y_0 + \lambda V, Z_0 + \lambda W) \text{ dove } -\infty < \lambda < +\infty$$

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

$$P_\lambda = \left( f \frac{X_0 + \lambda U}{Z_0 + \lambda W}, f \frac{Y_0 + \lambda V}{Z_0 + \lambda W} \right)$$

Per  $\lambda \rightarrow \pm\infty$  otteniamo  $P_\infty = \left( f \frac{U}{W}, f \frac{V}{W} \right)$ , se  $W \neq 0$ , detto punto di fuga associato alla famiglia di rette con direzione  $(U, V, W)$  e le linee che hanno la stessa direzione condividono il punto di fuga.



## Proiezione ortografica

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

In pratica se la **profondità**  $Z$  di tutti i punti dell'oggetto varia in un intervallo  $\Delta Z$  relativamente piccolo, allora il **fattore di scala prospettico**  $f/Z$  può essere approssimato da una costante  $s = f/Z_0$ . Le **equazioni di proiezione** diventano allora

$$X' = sX \text{ e } Y' = sY$$

con un **fattore di scaling** pari a  $f/Z_0$ .



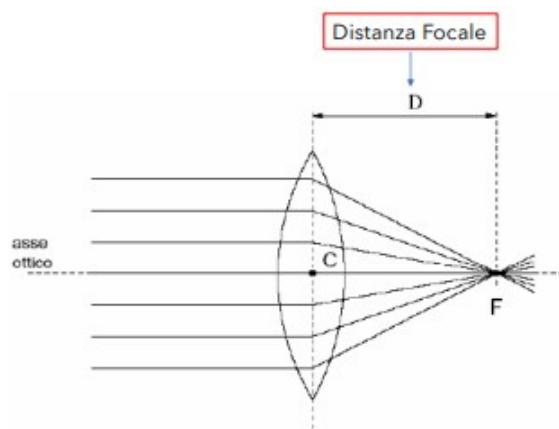
## Lenti sottili

Gli occhi, le macchine fotografiche e le telecamere utilizzano **lenti**, che essendo più grandi di un pinhole riescono a raccogliere più luce ma al contempo non può essere messa a fuoco tutta la scena contemporaneamente.

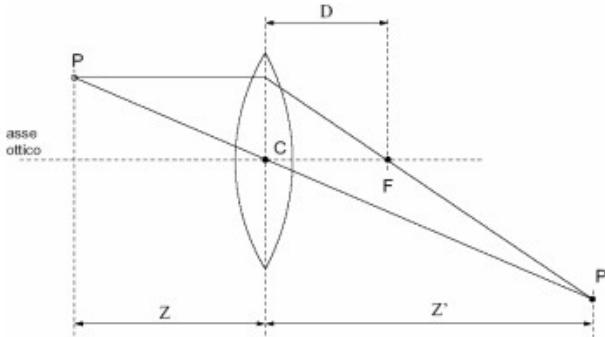
Come ottica dei sistemi di acquisizione terremo conto delle lenti sottili, sebbene questa sia più complessa in quanto composta da più lenti.

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 restano inalterati;
- la distanza del fuoco F dal centro C della lente prende il nome di **distanza focale D**, dipendente dai raggi di curvatura delle due superfici della lente e dall'indice di rifrazione del materiale.



Dato un punto P della scena, è possibile costruirne graficamente l'immagine  $P'$ , o **punto coniugato**, servendosi del raggio parallelo all'asse ottico che dopo la rifrazione passa per F e del raggio che passa inalterato per C.

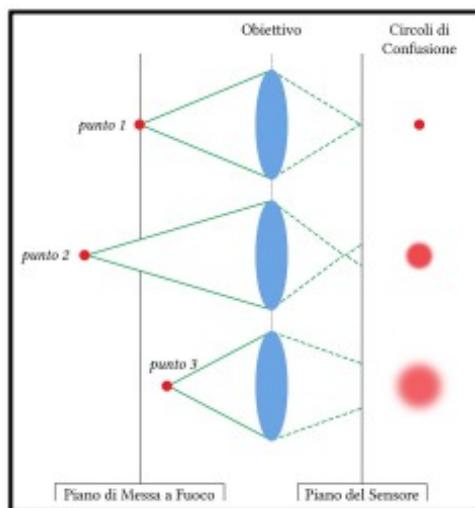


Dalla costruzione appena vista e dalla similarità dei triangoli otteniamo la **formula dei punti coniugati**, detta **equazione della lente sottile**:

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

In pratica l'immagine di un punto P della scena, distante Z dalla lente, viene prodotta a fuoco ad una distanza dalla lente Z' che dipende dalla profondità Z del punto e dalla distanza focale D della lente (Z rappresenta la distanza tra P e C e Z' rappresenta la distanza tra C e P').

Cambiando Z si ottiene l'**immagine sfocata** del punto, ossia il **cerchio di confusione**. Più un soggetto è lontano dal punto di messa a fuoco e più grandi saranno i cerchi che il soggetto forma sul sensore. La medesima cosa accade quando il soggetto è troppo vicino. Quando però un cerchio è sufficientemente piccolo, nella foto risulterà indistinguibile da un punto; questo genera una zona sufficientemente nitida chiamata **profondità di campo**.



Per mettere a fuoco oggetti a distanze diverse, le lenti dell'occhio cambiano forma, mentre le lenti delle telecamere traslano nella direzione Z.

La telecamera pinhole ha una profondità di campo infinita, mentre per le telecamere reali la profondità di campo è inversamente proporzionale al diametro della lente.

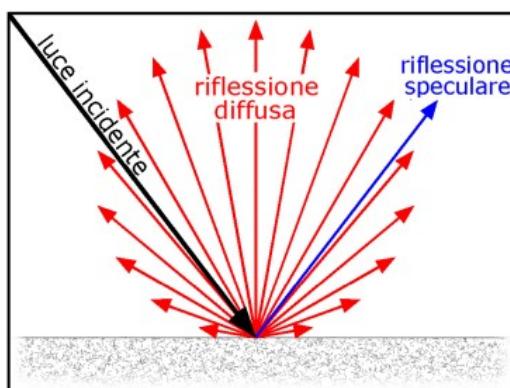
La distanza focale della lente e la distanza focale della telecamera pinhole sono cose diverse, nonostante condividano lo stesso nome.

## Fotometria della formazione dell'immagine

La **luminosità di un pixel** nell'immagine è funzione della luminosità della porzione di superficie proiettata nel pixel. Questa, a sua volta, dipende dall'**intensità complessiva della luce d'ambiente**, da se un **punto è illuminato o in ombra** e dalla **quantità di luce riflessa dal punto**.

La **riflessione** è un fenomeno complesso che dipende dalla superficie. I suoi **due casi estremi** sono:

- **riflessione diffusa**, nel caso in cui la luce venga diffusa in modo uniforme in **tutte le direzioni** in uscita dalla superficie;
- **riflessione speculare**, nel caso in cui la luce venga riflessa **in un'unica, o quasi, direzione**, con angolo di incidenza della luce uguale ad angolo di riflessione della stessa.



Ovviamente se la superficie non vede la luce, allora sarà in ombra.

Es.



### Specularità

#### Effetti dell'illuminazione

- Specularità sull'oliera in acciaio
- Le cipolle e le carote sono superfici a illuminazione diffusa perché sono rivoltate nella direzione della luce
- Le ombre appaiono in punti della superficie che non possono vedere la fonte di luce.
- All'interno della pentola ci sono alcune superfici scure diffuse dove la luce colpisce ad angolo tangenziale (anche delle ombre all'interno della pentola)

## Acquisizione delle immagini (Lez.6)

### Radiazioni elettromagnetiche

La radiazione elettromagnetica può essere considerata come un'onda o come un flusso di particelle prive di massa (fotoni).

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

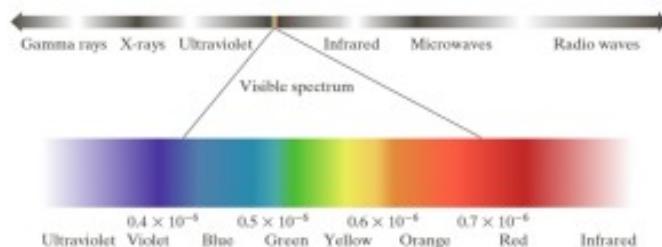
I colori che compongono lo spettro non sono ben separati, ma piuttosto variano gradualmente l'uno verso l'altro.

La luce priva di colore è detta monocromatica ed è caratterizzata solo dalla sua intensità, o livello di grigio.

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

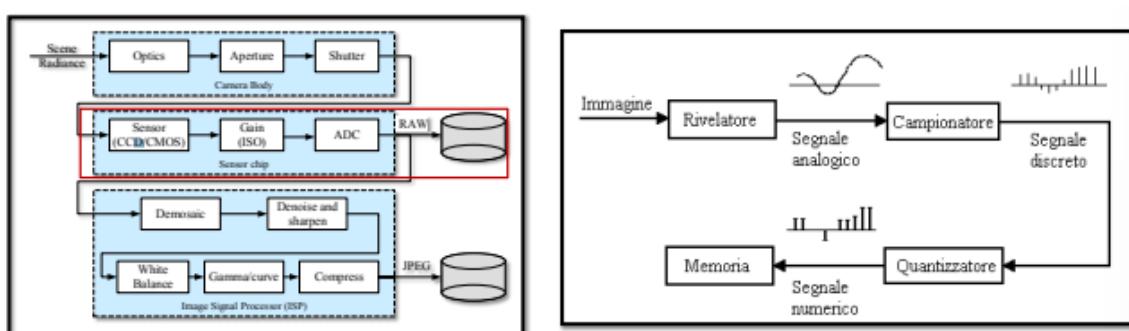
Oltre alla sua frequenza, o colore, la luce può essere caratterizzata da:

- **radianza**, ossia l'energia totale emessa dalla fonte di luce (si misura in watt - W);
- **luminanza**, cioè l'energia percepita dall'osservatore a partire da quella emessa dalla fonte di luce (si misura in lumen - lm);
- **luminosità**, che invece è una percezione soggettiva e di conseguenza non misurabile.



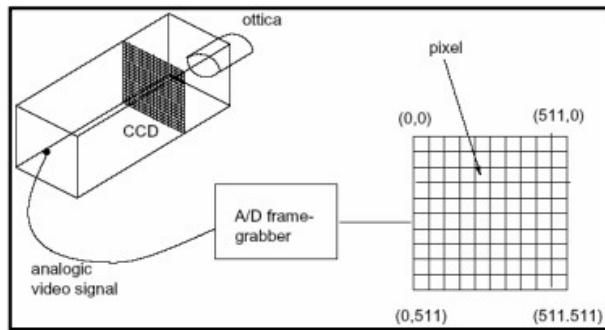
### Processo di acquisizione delle immagini

Il processo di conversione dei segnali analogici catturati da un dispositivo di acquisizione in segnali digitali è stato già trattato in segnali analogici e digitali e discretizzazione.



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

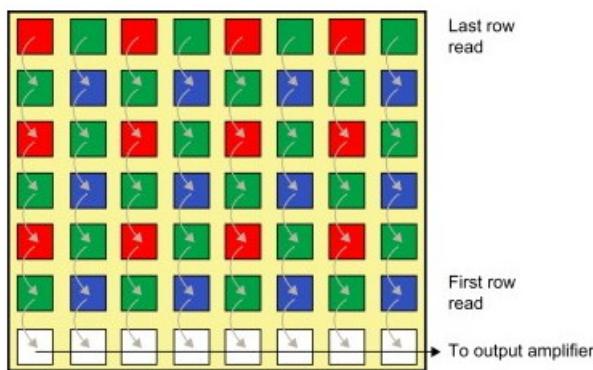
- una **telecamera**, composta dall'ottica che approssimiamo con una lente sottile e da una matrice di sensori, come può essere il **CCD (Charge-Coupled Device)**, che costituisce il piano immagine;
- un **frame grabber**, che è un dispositivo che si occupa di catturare e digitalizzare i segnali analogici provenienti da sorgenti esterne;
- un **calcolatore host**, che riceve i dati digitalizzati dal frame grabber e svolge operazioni di elaborazione, visualizzazione e archiviazione delle immagini.



**Il CCD 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 quando si usa un CCD è un segnale elettrico analogico**, ottenuto leggendo il potenziale degli elementi della matrice CCD per righe.

In un CCD, dopo l'esposizione, le cariche sulla prima riga vengono trasferite sul registro di lettura, dopodichè i segnali vengono inviati ad un amplificatore e poi ad un convertitore analogico-digitale. Ad ogni lettura, le cariche sul registro (di lettura) sono eliminate e si procede a far entrare la riga successiva. Le cariche su ogni riga, in pratica, vengono accoppiate a quelle sulla riga superiore, quindi quando la prima si sposta verso il basso, anche la successiva si sposta verso il basso e così via.



**I segnali analogici in uscita dal CCD vengono letti dal frame grabber che procede a digitalizzarli**, convertendoli in una matrice NxM (es. 512x512) di valori interi (es. 0-255), memorizzati in un'opportuna area di memoria (detta **frame**)

**buffer**), detti **pixel** (picture element). Il **calcolatore host** acquisisce poi l'immagine, trasferendola dal frame buffer alla memoria centrale.

La dimensione **NxM** della matrice CCD non è necessariamente la stessa dell'immagine **nxm** della matrice di pixel. Per questo motivo la posizione di un punto del piano immagine è diversa se misurata in elementi CCD piuttosto che in pixel

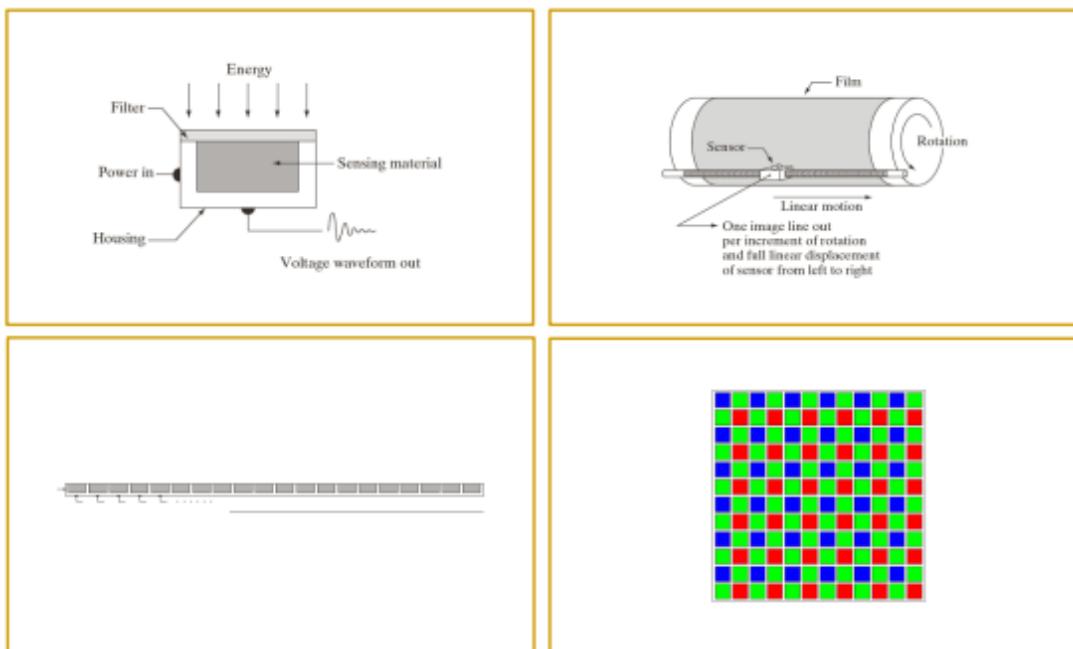
$$u_{pix} = \frac{N}{n} u_{CCD} \quad e \quad v_{pix} = \frac{M}{m} v_{CCD}$$

ma è comodo comunque assumere che vi sia una relazione uno ad uno tra le due.

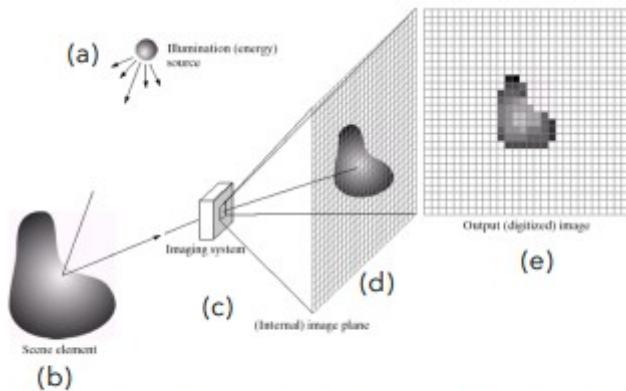
## Dispositivi di acquisizione

Per scansionare una scena è possibile utilizzare i sensori in diverse modalità:

- **singoli sensori**, che fanno uso di congegni meccanici per spostare il sensore rispetto alla scena;
- **sensori disposti in linea**, che fanno uso del movimento di un mezzo per ottenere la seconda dimensione;
- **array 2D di sensori**, che richiedono ulteriori elaborazioni per ottenere la sezione dell'oggetto scansionato dai dati acquisiti.



## Esempio di acquisizione delle immagini



La radiazione (a) è riflessa dall'oggetto (b) e catturata dal sistema di imaging (c). La scena poi è proiettata su un piano immagine formato da sensori (d) ed è digitalizzata (e)

## Modello di formazione dell'immagine

Ricordiamo che un'immagine è definita come una **funzione bidimensionale**  $f(x,y)$ , dove  $x$  e  $y$  sono le coordinate spaziali (nel piano) e l'ampiezza di  $f$  in corrispondenza di 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 0, cioè  $0 < f(x,y) < \infty$ .

La funzione  $f(x,y)$  è caratterizzata da:

- **illuminazione**, che è la quantità di luce incidente sulla scena ed è anch'essa finita e diversa da 0,  
$$0 < i(x,y) < \infty;$$
- **riflettanza**, ossia la quantità di luce riflessa dagli oggetti presenti nella scena ed è compresa tra 0, totale assorbimento, e 1, totale riflessione,  
$$0 < r(x,y) < 1.$$

Avremo dunque che  $f(x,y) = i(x,y) * r(x,y)$ .

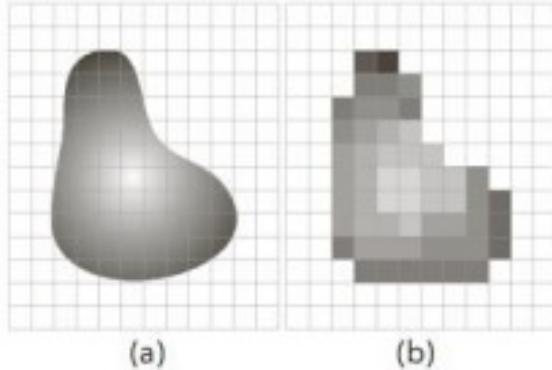
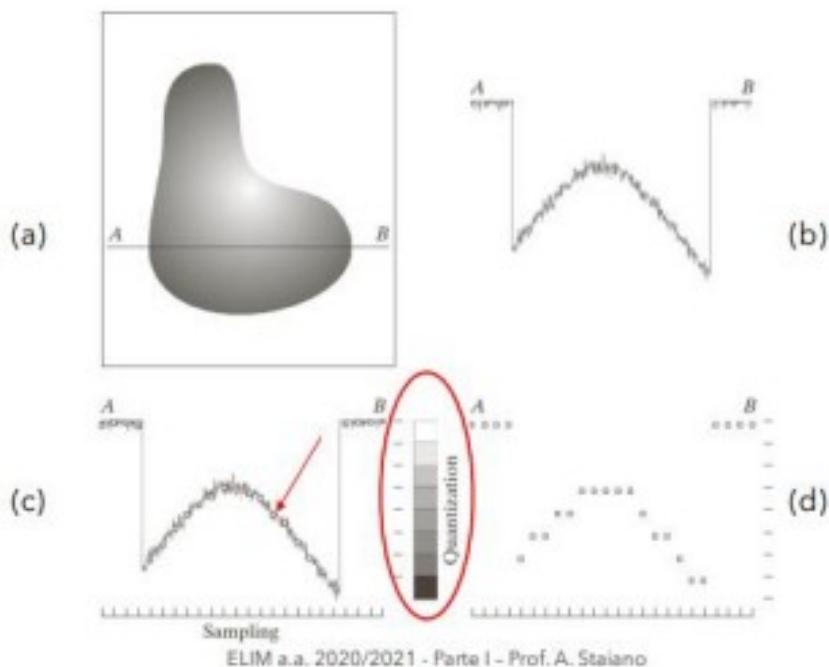
Nella pratica l'intensità  $f(x,y)$  è compresa in una **scala di grigio**  $[L_{min}, L_{max}]$ , con  $L_{min} = i_{min} r_{min}$  e  $L_{max} = i_{max} r_{max}$ , **convenzionalmente scalata in [0, L-1]**, ossia [nero, bianco].

## Campionamento e quantizzazione

Innanzitutto ricordiamo cosa sono campionamento e quantizzazione.

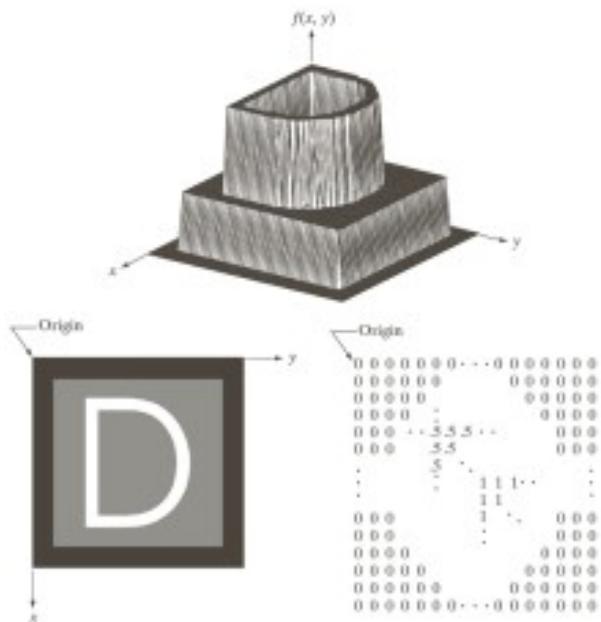
In generale l'immagine osservata dal dispositivo di acquisizione viene proiettata sull'**array di sensori**, dove sarà sottoposta alla **fase di discretizzazione**, ossia campionamento e quantizzazione.

Il livello di grigio di ogni pixel dell'immagine è ottenuto come **valore medio della regione corrispondente nel campionamento**, approssimato al livello di grigio più vicino tra quelli disponibili.



## Rappresentazione delle immagini

L'immagine  $f(x,y)$  viene rappresentata come una matrice  $M \times N$  ad **L valori discreti**. L è generalmente una potenza di 2, quindi  $L = 2^k$ , in modo che ogni pixel venga rappresentato utilizzando k bit.



Questo è il tipo di rappresentazione delle immagini più utilizzato, ma è possibile farlo anche attraverso equazioni.

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

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

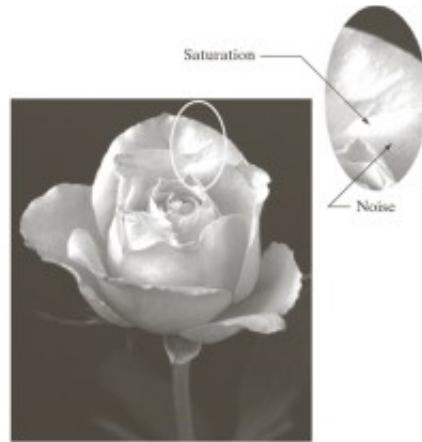
**Il sensore esegue una misurazione dell'intensità di luce**, ma poichè è una misura strumentale, il sensore **non è privo di errori (limitazione del campionamento)**.

**Il range dinamico**, ossia l'**insieme di valori misurati nella scala di grigio**, di un sistema di imaging è il rapporto tra l'intensità massima misurabile e il minimo livello di intensità rilevabile nel sistema.

Essendo il **campionamento limitato**, possono crearsi fenomeni come:

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

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



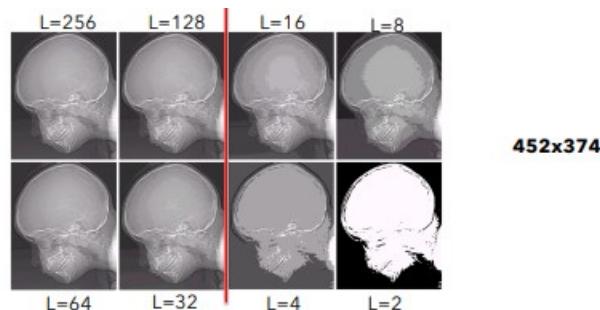
Il numero di bit  $b$  richiesto per rappresentare un'immagine  $M \times N$  a  $L$  livelli di grigio è

$$b = MN \log_2 L, \text{ che per } L=2^k, b = MNk.$$

## Risoluzione spaziale e dell'intensità

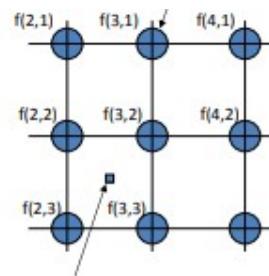
La **risoluzione spaziale di un'immagine** è la dimensione del più piccolo dettaglio che può essere riconosciuto nell'immagine ed è spesso misurata in dpi (dot per inch – punti per pollice).

La **risoluzione dell'intensità (o del contrasto)**, invece, 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 di immagini

L'interpolazione di immagini è il **processo mediante il quale si utilizzano dei dati noti per stimare i valori di un'immagine in posizioni non intere**.

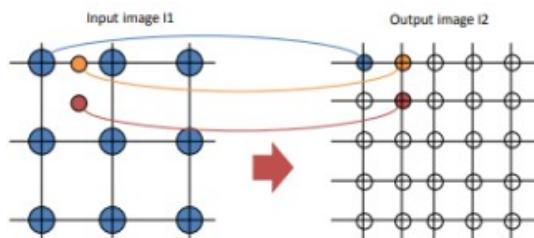


Qual è il valore qui, in  $f(2.5, 2.5)$ ?

### Abbiamo 3 tipi di interpolazione:

- **nearest neighbour**, nella quale si prende il **valore del dato più vicino**;
- **bilineare**, dove si prende una **combinazione dei 4 più vicini**, ed equivale a fare l'interpolazione prima in una dimensione e poi nell'altra;
- **cubica**, nella quale si prendono i **valori dei 16 più vicini** (più costosa ma con risultati migliori).

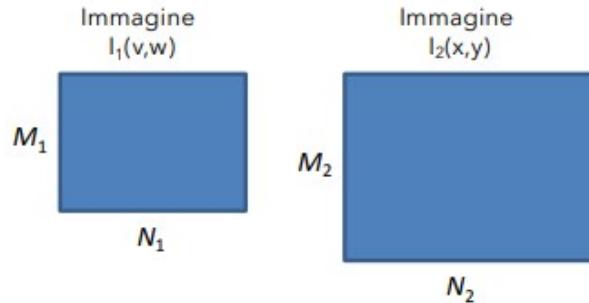
In generale si parte da una matrice che avrà un valore in ogni posizione intera. Nel caso in cui volessimo effettuare uno **zoom**, avremo come input l'immagine originale e come output quella scalata. Per assegnare i valori ai pixel dell'immagine in output, li scandiamo (i pixel), calcolando per ognuno il valore a partire dalla posizione corrispondente dell'immagine originale. Se la posizione nell'immagine di input non è intera, allora procediamo con l'interpolazione.



### Funzione di mapping

Per effettuare operazioni di ingrandimento o rimpicciolimento di un'immagine si ha bisogno anche di una funzione di mapping, come quella riportata di seguito.

- Assumiamo
  - Immagine  $I_1$  di dimensioni  $M_1 \times N_1$
  - Immagine  $I_2$  di dimensioni  $M_2 \times N_2$
- Allora
  - $x = c_x \cdot v$
  - $y = c_y \cdot w$
  - 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
    - $v = x / c_x$
    - $w = y / c_y$



### Relazioni tra pixel

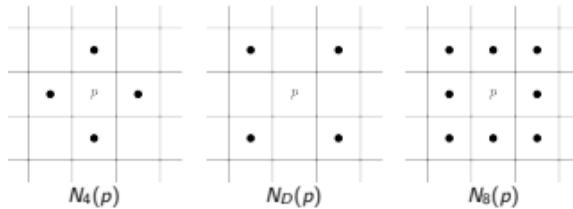
La struttura di un'immagine digitale permette di definire alcune relazioni basilari tra i pixel, che possono essere utili in vari casi pratici. In particolare alcune operazioni considerano **gruppi di pixel che condividono le stesse caratteristiche**.

Le relazioni possono riguardare la **vicinanza**, l'**adiacenza**, la **connettività** e la **misura della distanza sul dominio dell'immagine**.

## Vicinanza

La relazione di vicinanza, come si può immaginare, è definita considerando i **pixel "vicini"**. Ne distinguiamo **3 tipi**:

- $N_4$ , la **4-vicinanza**, dove i pixel vicini sarebbero quelli nelle posizioni sopra, sotto, a destra e a sinistra del pixel considerato, ossia  $(x+1,y)$ ,  $(x-1,y)$   $(x,y+1)$  e  $(x,y-1)$ ;
- $N_D$ , la **vicinanza diagonale**, dove i pixel vicini sarebbero quelli nelle posizioni sopra a destra, sotto a destra, sopra a sinistra e sotto a sinistra del pixel considerato, ossia  $(x+1,y+1)$ ,  $(x-1,y+1)$   $(x+1,y-1)$  e  $(x-1,y-1)$ ;
- $N_8$ , la **8-vicinanza**, dove i pixel vicini sono dati dall'unione della 4-vicinanza con quelli della vicinanza diagonale.



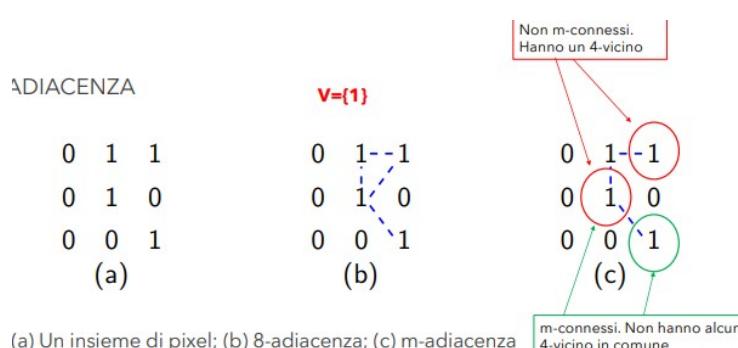
## Adiacenza

La relazione di adiacenza, invece, è definita considerando i **pixel "simili"**.

In questo contesto viene utilizzata l'**intensità dei pixel** per definire la relazione di similarità, infatti si ha un insieme  $V$  composto dalle intensità che devono essere considerate nella definizione di adiacenza.

Considerando i pixel  $p$  e  $q$ , che hanno valori di intensità in  $V$ , questi sono:

- **4-adiacenti** se  $q$  è 4-vicino di  $p$ ;
- **8-adiacenti** se  $q$  è 8-vicino di  $p$ ;
- **m-adiacenti**, con la  $m$  che sta per **mista** (quindi adiacenza mista), se  $q$  è 4-vicino di  $p$  oppure se è vicino diagonale di  $p$  con nessun 4-vicino in comune.



Il percorso (path) dal pixel p con coordinate  $(x_p, y_p)$  al pixel q di coordinate  $(x_q, y_q)$  è una sequenza di  $n+1$  pixel distinti di 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, con  $1 \leq i \leq n$ .

La definizione del percorso dipende dalla definizione di adiacenza, quindi si parlerà di 4-percorso, 8-percorso o m-percorso, ed inoltre se  $(x_0, y_0) = (x_n, y_n)$  allora si tratta di un **percorso chiuso**.

## 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 loro formato interamente da pixel appartenenti ad S.

Per ciascun pixel p in S, l'insieme di pixel ad esso connessi costituisce una **componente connessa di S**.

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

E' invece chiamato **regione** dell'immagine R un qualsiasi sottoinsieme connesso.

0	1	1	0	0	0	0	0	0	3	3	3	0	0	0	0	0
0	0	1	1	0	0	0	0	0	3	3	3	3	3	3	3	3
1	1	1	0	0	0	0	0	0	0	3	3	3	3	0	0	0
0	0	1	1	0	0	2	2	2	0	0	0	3	3	3	3	3
0	0	1	1	1	1	2	2	2	2	0	3	3	3	0	0	0
0	0	0	1	1	0	0	2	2	2	0	0	3	3	3	0	0

Usando la 4-adiacenza

Oss: ci sono due regioni per 3

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

- due regioni  $R_i$  e  $R_j$  sono adiacenti se la loro unione 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**, mentre il suo complemento, ossia l'insieme dei pixel che non appartengono ad alcuna regione, prende il nome di **background**.

Spesso gli oggetti in foreground vengono circondati da pixel di background aggiuntivi, ad esempio per mettere in risalto il soggetto dell'immagine.

Un altro concetto importante è il bordo, il quale dipende dall'adiacenza.

Il **bordo di una regione** R è l'insieme dei pixel di R adiacenti al complemento di R, cioè il bordo di una regione è composto da pixel che hanno almeno uno dei propri vicini nel background.

0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Il pixel cerchiato appartiene al bordo  
solo se è usata la 8-adiacenza

Questo, in realtà, prende il nome di **bordo interno**, per distinguerlo dal **bordo esterno**, che invece è composto da pixel di background che sono adiacenti al bordo interno.

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

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

Il concetto di bordo non deve essere confuso con quello di **edge** (lato), che è invece una sottosequenza del bordo.

## Distanza

La funzione di distanza è una funzione D definita su coppie di pixel tali che:

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

Esistono **3 tipi di distanze**:

- **distanza Euclidea**  $D_e$  (norma  $L_2$ ), che è data da

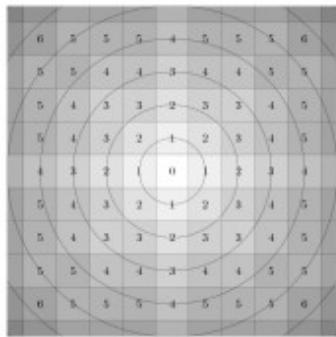
$$D_e(p, q) = \left[ (x_p - x_q)^2 + (y_p - y_q)^2 \right]^{\frac{1}{2}}$$

- **distanza**  $D_4$  (norma  $L_1$ ), anche detta **city-block**, che è data da

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

- **distanza**  $D_8$  (norma  $L_\infty$ ), anche detta **scacchiera**, che è data da

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



$D_e$

8	7	6	5	4	5	6	7	8
7	6	5	4	3	4	5	6	7
6	5	4	3	2	3	4	5	6
5	4	3	2	1	2	3	4	5
4	3	2	1	0	1	2	3	4
5	4	3	2	1	2	3	4	5
6	5	4	3	2	3	4	5	6
7	6	5	4	3	4	5	6	7
8	7	6	5	4	5	6	7	8

$D_4$

4	4	4	4	4	4	4	4	4
6	3	3	3	3	3	3	3	3
4	3	2	2	2	2	2	3	4
4	3	2	1	1	1	2	3	4
6	3	2	1	0	1	2	3	4
4	3	2	1	1	1	2	3	4
4	3	2	2	2	2	2	3	4
6	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4

$D_8$

La distanza può essere valutata anche utilizzando la m-adiacenza, ed in questo caso è definita come l'**m-path più breve tra i punti**, quindi dipenderà dal valore dei pixel lungo il path.

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

## Operazioni sulle immagini (Lez.7)

### Operazioni puntuale e matriciali

Come sappiamo, le immagini sono rappresentate da matrici.

Un'operazione puntuale, su una o più immagini, viene effettuata pixel-wise (ossia pixel per pixel).

Il **prodotto puntuale** di due immagini è definito come

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

mentre il **prodotto matriciale** è definito righe per colonne, ossia

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

### Operazioni aritmetiche

Un tipo di operazione puntuale sono le operazioni aritmetiche, per cui date due immagini  $f(x,y)$  e  $g(x,y)$  di uguali dimensioni  $M \times N$ , si ha:

- Addizione:  $a(x,y) = f(x,y) + g(x,y)$
- Sottrazione:  $s(x,y) = f(x,y) - g(x,y)$
- Moltiplicazione:  $p(x,y) = f(x,y) \times g(x,y)$
- Divisione:  $d(x,y) = f(x,y) / g(x,y)$

Questi tipi di operazioni vengono utilizzati in diversi campi, ad esempio la **somma** viene utilizzata per **ridurre il rumore delle immagini**, infatti addizionando un'immagine con se stessa avremo un'immagine con meno rumore di quella iniziale. Un altro esempio di applicazione è quello della **sottrazione**, che permette il **confronto tra immagini** verificando la variazione tra i pixel delle due immagini.

## Operazioni lineari e non lineari

Consideriamo un **operatore H** che, data un'immagine di input  $f(x,y)$ , fornisca in output un'immagine  $g(x,y)$  data da  $H[f(x,y)] = g(x,y)$ . Considerando inoltre due costanti arbitrarie  $a$  e  $b$ ,  $H$  è detto operatore lineare se rispetta l'equazione

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

altrimenti è detto non lineare. In pratica questa equazione afferma che applicare l'operatore  $H$  alla combinazione lineare di due funzioni (immagini) equivale ad applicare l'operatore alle singole combinazioni lineari.

**Un esempio** di operazione lineare è la **somma** (prima immagine sottostante), mentre al contrario **non è un'operatore lineare il massimo** (max, seconda immagine sottostante).

Esempio (operatore lineare):

$$\Sigma[af_1(x,y) + bf_2(x,y)] = \Sigma af_1(x,y) + \Sigma bf_2(x,y) = a \Sigma f_1(x,y) + b \Sigma f_2(x,y) = a g_1(x,y) + b g_2(x,y)$$

- Date le immagini  $\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}$  e  $\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$  ed inoltre le costanti  $a = 1$  e  $b = -1$ , allora applicando la (1) si ha
- $\max \left\{ (1) \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1) \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = \max \left\{ \begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix} \right\} = -2$   
tuttavia
- $\max \left\{ \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \right\} + (-1) \max \left\{ \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = 3 + (-1)7 = -4$

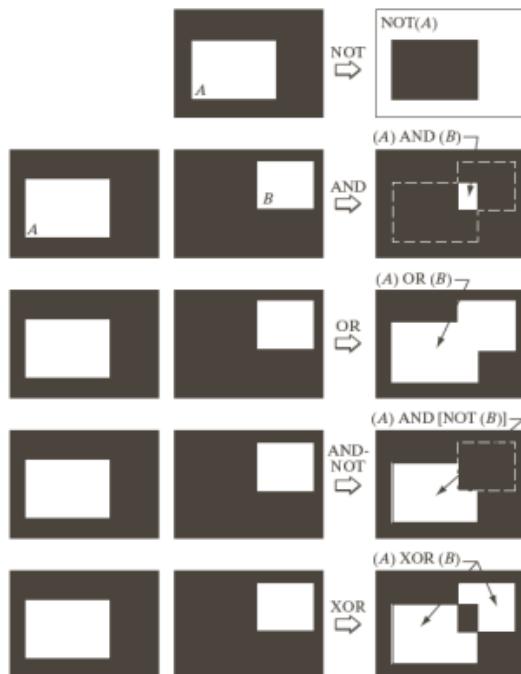
## Operazioni insiemistiche e logiche

Le operazioni insiemistiche e logiche trovano la loro applicazione nell'ambito delle **immagini binarie** (bianco e nero).

Nell'elaborazione delle immagini gli **insiemi** sono oggetti (regioni) in un'immagine binaria, di conseguenza se vogliamo ad esempio sapere se due oggetti A e B si sovrappongono, basta calcolare l'intersezione tra i due.

Description	Expressions
Operations between the sample space and null sets	$\Omega' = \emptyset; \emptyset' = \Omega; \Omega \cup \emptyset = \Omega; \Omega \cap \emptyset = \emptyset$
Union and intersection with the null and sample space sets	$A \cup \emptyset = A; A \cap \emptyset = \emptyset; A \cup \Omega = \Omega; A \cap \Omega = A$
Union and intersection of a set with itself	$A \cup A = A; A \cap A = A$
Union and intersection of a set with its complement	$A \cup A' = \Omega; A \cap A' = \emptyset$
Commutative laws	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distributive laws	$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
DeMorgan's laws	$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$

Per quanto riguarda gli **operatori logici**, avremo AND, OR, NOT, NAND e XOR.



Le operazioni insiemistiche, ma anche quelle logiche, **possono essere utilizzate sia considerando le coordinate di singole regioni di pixel di foreground come insiemi in una singola immagine, che considerando una o più immagini della stessa dimensione**, in tal caso eseguendo le operazioni tra pixel corrispondenti.

## Operazioni spaziali

Le operazioni spaziali **sono eseguite direttamente sui pixel di un'immagine** e ne esistono di **3 categorie**:

- operazioni su singoli pixel;
- operazioni di vicinanza;
- trasformazioni spaziali geometriche.

## Operazioni su singoli pixel

Nelle operazioni su singoli pixel si altera l'intensità degli stessi (singoli pixel) utilizzando una **funzione di trasformazione**

$$s = T(z),$$

con z che è l'intensità di un pixel nell'immagine originale e s che è l'intensità corrispondente nell'immagine di output.

## Operazioni di vicinanza

Quando le **elaborazioni coinvolgono i valori dei pixel in un piccolo intorno di una posizione (x,y) di interesse**, allora si parla di operazioni di vicinanza.

In questo tipo di operazioni, oltre ad una porzione di immagine, vengono utilizzate anche le cosiddette **maschere**. In pratica **per calcolare il valore del pixel al centro di una porzione**, bisogna "centrare" la maschera su di esso, moltiplicare ogni pixel della porzione (intorno) per il corrispondente **coefficiente** (peso) della maschera, ed infine sommare i risultati.

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

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

Di conseguenza si può ben capire che il risultato di un'operazione di questo tipo dipende dal livello di grigio nel punto che vogliamo calcolare, dal livello di grigio dei pixel nel suo intorno e dal valore dei coefficienti della maschera.

Le operazioni di vicinanza sono molto utilizzate, soprattutto in applicazioni come la **riduzione del rumore**, ma **applicare una maschera a tutti i pixel** di un'immagine risulta **molto costoso**. Di conseguenza oltre certe soglie vengono preferite tecniche diverse.

## Trasformazioni spaziali geometriche

Le trasformazioni spaziali geometriche vengono utilizzate per **modificare la disposizione spaziale dei pixel** in un'immagine e **consistono di due operazioni base**:

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

La **trasformazione delle coordinate** può essere espressa come

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

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

Un particolare tipo di trasformazioni spaziali geometriche sono le **trasformazioni affini**, dette così poiché preservano rette, punti e piano. Abbiamo quindi:

- **scaling;**
- **traslazione;**
- **rotazione;**
- **distorsione (shearing).**

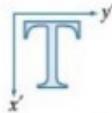
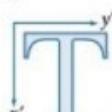
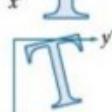
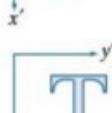
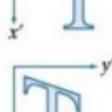
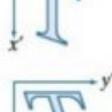
Si rappresentano tutte con l'equazione appena vista, tranne la traslazione che richiede l'aggiunta di un vettore 2D costante al membro destro dell'equazione.

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

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

In questo caso si parla di **rappresentazione con coordinate omogenee**, che è molto utile in quanto permette di concatenare una serie di trasformazioni.

*Trasformazioni affini in coordinate omogenee*

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

Tutte le trasformazioni appena viste spostano le coordinate dei pixel di un'immagine in nuove posizioni, di conseguenza è necessario assegnare i valori di intensità in queste posizioni, mediante un processo di **interpolazione**.

In particolare la matrice 3x3 che esprime le trasformazioni affini può essere utilizzata con:

- **mapping diretto** (forward mapping), se procede alla scansione dei pixel dell'immagine di input e ad ogni posizione  $(x,y)$  calcola la nuova posizione  $(x',y')$  del corrispondente pixel nell'immagine di output. Questa soluzione però **può portare a problemi**, infatti due o più pixel dell'immagine di input possono essere trasformati nella stessa posizione dell'immagine di output o anche alcune posizioni in output potrebbero non corrispondere ad alcun pixel;
- **mapping inverso**, se procede alla scansione dei pixel di output e ad ogni posizione  $(x',y')$  calcola la posizione corrispondente nell'immagine di input con  $(x,y) = A^{-1}(x',y')$ , effettuando successivamente l'**interpolazione** utilizzando i pixel di input più vicini. Questa soluzione risulta essere **più efficiente** rispetto a quella del mapping diretto.

## Trasformazioni di intensità (Lez.8)

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, con le **tecniche del dominio della frequenza** (che permette di rappresentare le caratteristiche di un segnale in termini di frequenze e ampiezze delle sue componenti). Tra le due, le tecniche del dominio spaziale risultano essere meno gravose dal punto di vista computazionale.

Formalmente, per le tecniche del dominio spaziale, possiamo dire che

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

ossia che un'immagine, o sottoimmagine, di output sarà uguale alla **trasformazione** applicata sull'immagine, o sottoimmagine, di input.

L'operatore **T** è di solito **definito su un'opportuna area di vicinanza di  $(x,y)$** , ma quando essa (la vicinanza) fuoriesce dall'immagine, devono essere utilizzati alcuni **criteri di estensione**, come ad esempio il **padding**.

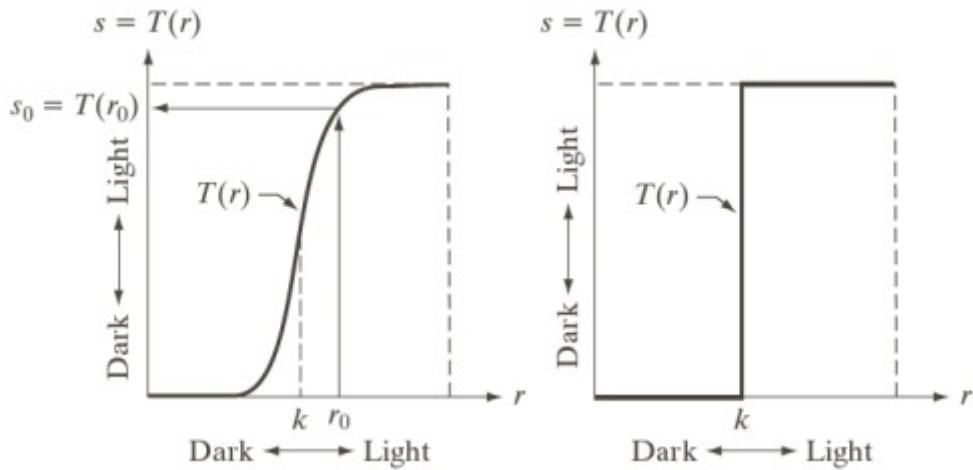
Se il raggio della vicinanza è 0, la trasformazione coinvolge solo il pixel considerato e dipende solo dalla sua intensità. In questo caso, quindi, **g dipende solo dal valore di f in un singolo punto  $(x,y)$**  e T diventa una funzione di trasformazione dell'intensità, la cui formula è

$$s = T(r),$$

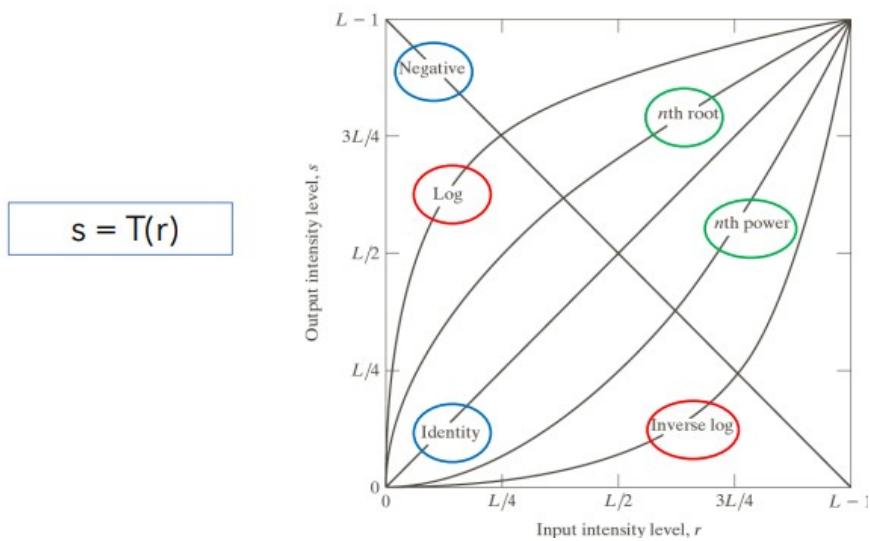
con  $s$  e  $r$  che denotano rispettivamente l'intensità di  $g$  e di  $f$ .

Le tecniche di trasformazione di intensità sono anche chiamate **elaborazioni puntuali**, sono semplici da implementare e vengono utilizzate per migliorare le immagini, a partire dalle richieste dell'utente e dal tipo di applicazione.

Es. di binarizzazione di un'immagine, dove tutti i pixel con intensità sotto una certa soglia vengono messi a nero, mentre quelli con intensità superiore vengono messi a bianco.



Si possono distinguere **diversi tipi di trasformazioni di intensità**, che vedremo nei seguenti paragrafi.

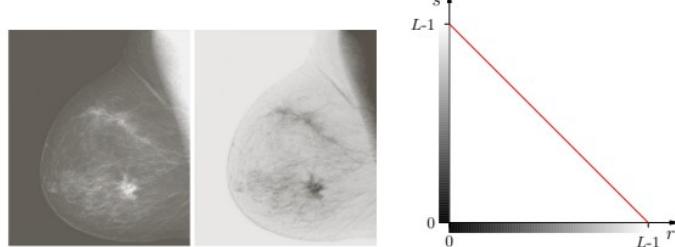


## Negativa di un'immagine

Certe volte i dettagli sono più evidenziabili quando l'intensità dei pixel è **invertita**, ad es. quando i dettagli sono di un bianco o grigio chiaro e il background è scuro e copre gran parte dell'immagine.

Si può quindi ottenere la negativa di un'immagine, con livelli di intensità nella gamma [0, L-1], utilizzando la **trasformazione**

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



## Trasformazioni logaritmiche

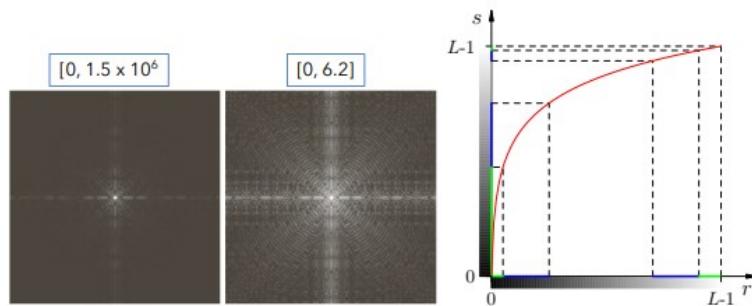
Le trasformazioni logaritmiche vengono **utilizzate per rappresentare funzioni bidimensionali che sono definite su ampi intervalli ed hanno picchi grandi e piccoli.**

Ad esempio consideriamo di avere un'immagine prevalentemente scura in cui però ci sono alcuni pixel estremamente chiari. Tramite l'utilizzo di una trasformazione logaritmica si possono rendere i pixel scuri più visibili, consentendo una visualizzazione più chiara dei dettagli.

La **forma generale** delle trasformazioni logaritmiche è

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

Es.



## Trasformazioni gamma

Le trasformazioni gamma, anche chiamate di potenza, **vengono utilizzate per correggere l'output del dispositivo di visualizzazione o anche per correggere il contrasto.**

La **forma generale** di queste trasformazioni è

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

ma viene talvolta riportata come

$$s = c(r + \varepsilon)^\gamma.$$

Il fattore di correzione  $\gamma$  è quello che ha maggior peso nell'equazione, ed in particolar modo l'immagine di output, rispetto all'immagine di input:

- risulterà **identica**, se  $\gamma = 1$ ;
- risulterà **schiarita**, se  $\gamma < 1$ ;
- risulterà **più scura**, se  $\gamma > 1$ .

Bisogna inoltre tenere conto del fatto che un valore troppo grande, o troppo piccolo, di  $\gamma$  potrebbe compromettere i risultati.

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)  $c = 1, \gamma = 5.0$

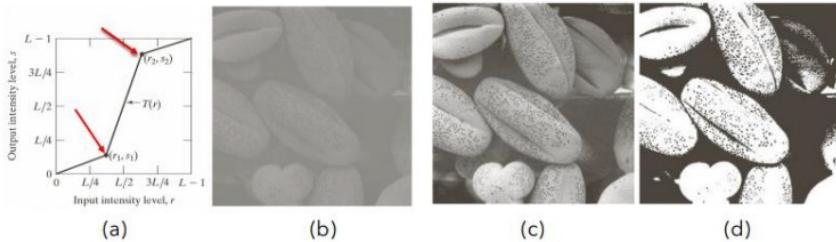
## Trasformazioni di stretching del contrasto

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.

Per fare ciò è **necessario selezionare dei punti**, e di conseguenza un intervallo, su cui si andrà ad effettuare l'ampliamento.

La forma della funzione di trasformazione dipende quindi dalle posizioni dei punti selezionati.

Di seguito abbiamo come esempio un'immagine a basso contrasto che viene elaborata ponendo un punto  $(r_1, s_1) = (r_{min}, 0)$  e un altro punto  $(r_2, s_2) = (r_{max}, L-1)$ , ottenendo un'immagine decisamente migliorata. Nel caso in cui  $r_1=s_1=0$  e  $s_2=L-1$ , la trasformazione diventa una **funzione soglia** che crea un'immagine binaria.

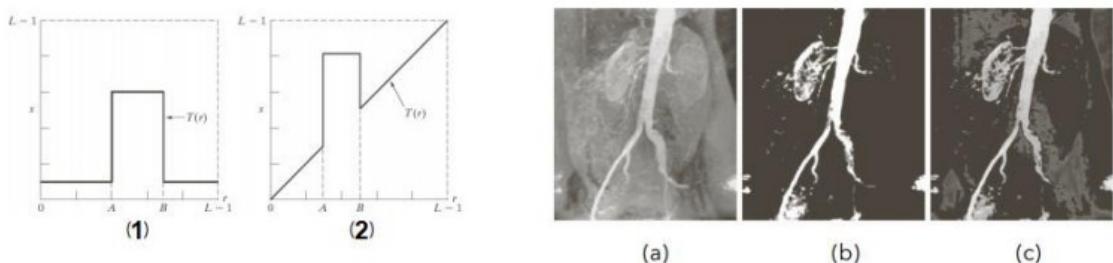


## Trasformazioni di selezione (slicing) del livello di intensità

Le trasformazioni di slicing del livello di intensità hanno lo scopo di **evidenziare alcune caratteristiche particolari di un'immagine, prendendo in considerazione solo un range di intensità.**

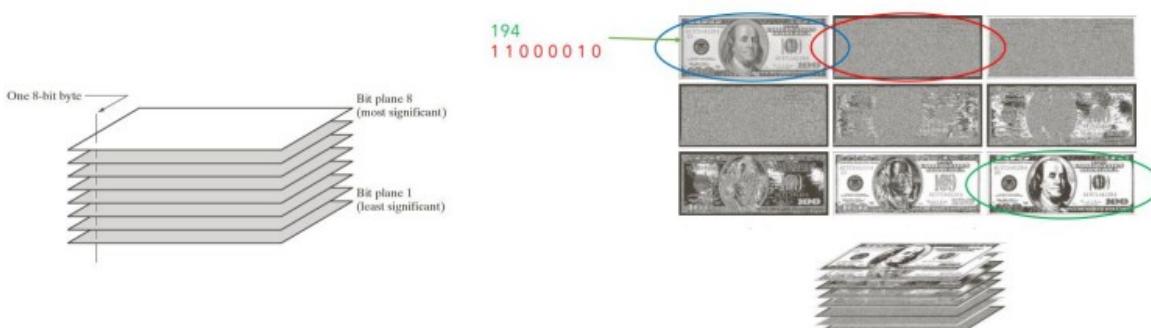
In questo tipo di trasformazioni si possono seguire **due approcci:**

- il primo consiste nell'**impostare tutte le intensità che non fanno parte del range ad un valore basso**, il che porta a schiarire le intensità facenti parte del range, scurendo al contempo tutto il resto;
- il secondo consiste nel **preservare le intensità che non fanno parte del range**, il che porta a schiarire le intensità facenti parte del range, mantenendo appunto inalterato tutto il resto.



## Trasformazioni bit-plane

Nelle trasformazioni bit-plane, l'immagine viene vista come composta da strati di bit, invece che da una matrice di interi. Ciò comporta che un'immagine ad esempio a 8 bit avrà 8 strati, con il piano bit 8 che sarà quello più significativo. **Ogni strato contribuisce all'aspetto finale dell'immagine, ma il più dell'informazione si trova negli strati più alti.**



## Trasformazioni basate su istogrammi

L'istogramma di un'immagine a L valori è una funzione discreta

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

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

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

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

con  $M$  e  $N$  ovviamente il numero di righe e colonne dell'immagine e  $p(k)$  una stima della probabilità di occorrenza di  $k$ .

L'istogramma fornisce uno strumento visuale intuitivo per valutare alcune proprietà statistiche dell'immagine, e le trasformazioni basate su istogramma possono essere di miglioramento, di compressione o di segmentazione.

Nel caso in cui si debba rappresentare un'**immagine scura** mediante **istogramma**, i **componenti** di quest'ultimo (istogramma) saranno **localizzati a bassi valori di intensità**, mentre accade l'**esatto opposto** nel caso in cui si debba rappresentare un'**immagine chiara**.

Se invece si devono rappresentare **immagini a basso contrasto**, i componenti dell'istogramma saranno **localizzati in una regione ristretta di valori di intensità**, mentre al contrario, per **immagini ad alto contrasto**, i componenti dell'istogramma saranno **quasi uniformemente distribuiti su tutto il range di intensità**, con pochi picchi.

L'obiettivo delle trasformazioni basate su **istogrammi** è proprio quello di raggiungere una distribuzione uniforme, in modo che l'immagine abbia un **elevato range dinamico**, così che i **dettagli** vengano percepiti più facilmente.

IMMAGINE SCURA

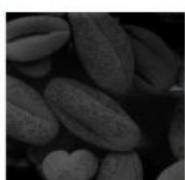


IMMAGINE CHIARA



IMMAGINE A BASSO CONTRASTO

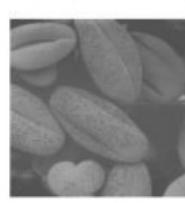
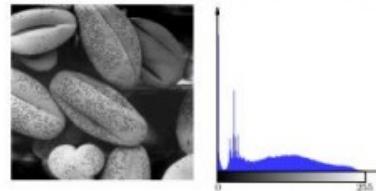


IMMAGINE AD ALTO CONTRASTO



Per studiare le trasformazioni basate su istogrammi bisogna prima introdurre il concetto di **trasformazioni monotoniche** (continue) su  $[0, L-1]$ .

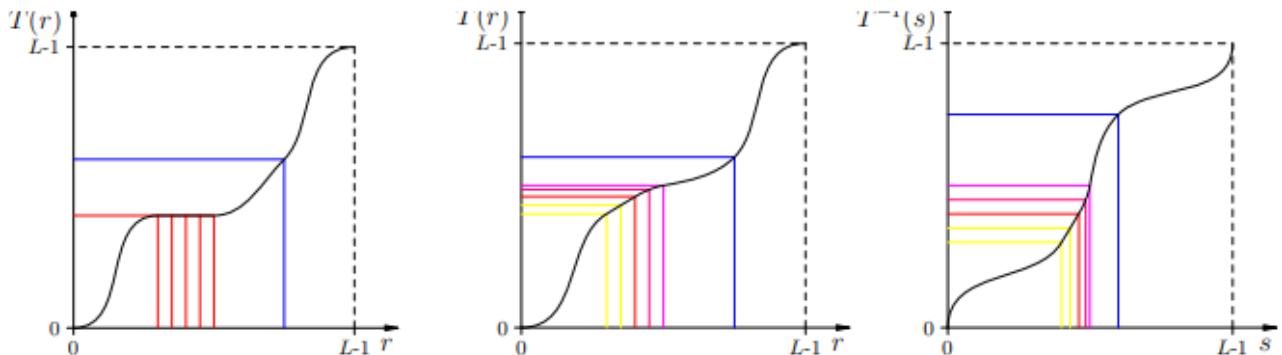
Supponiamo quindi di avere

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

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

Inoltre se  $T$  è strettamente monotona crescente, ossia se  $T(r_2) > T(r_1)$ , esiste  $T^{-1}$  tale che

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



## Equalizzazione e specifica dell'istogramma

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:

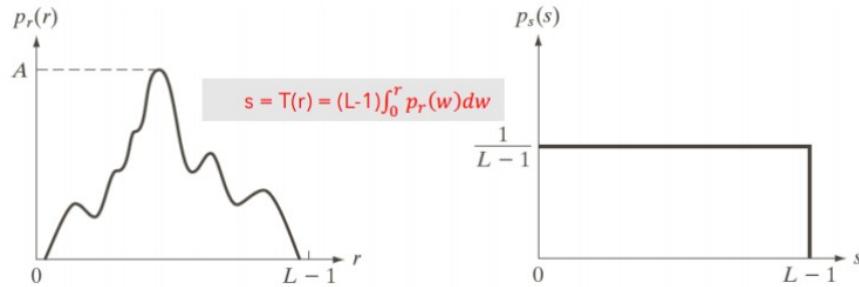
$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1) \frac{d}{dr} \left[ \int_0^r p_r(w) dw \right] = (L-1)p_r(r)$$

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  $p_r$ ,

con  $p_s(s)$  e  $p_r(r)$  che rappresentano rispettivamente la densità di probabilità delle intensità dei pixel nell'immagine trasformata e la densità di probabilità delle intensità dei pixel nell'immagine originale. Con **densità di probabilità** si intende la probabilità che un pixel abbia un certo livello di intensità.



Considerando  $r_k$  come il livello di intensità in  $[0, L-1]$ ,  $p_r(r_k) = \frac{n_k}{MN}$ , con  $k = 0, 1, \dots, L-1$

1. Un'immagine può quindi essere **equalizzata** assegnando l'intensità  $s_k$  a quei pixel che hanno intensità  $r_k$  tramite

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

considerando che gli istogrammi di immagini sono **funzioni discrete** e quindi sono rappresentati da un insieme finito di valori discreti, anziché da una distribuzione continua. In questo tipo di situazione la sommatoria viene preferita all'integrazione (integrali), poiché quest'ultima è tipica delle funzioni continue.

Questa trasformazione è detta **equalizzazione di istogramma**, che è una procedura base che permette di ottenere un'immagine con una distribuzione di intensità approssimativamente uniforme.

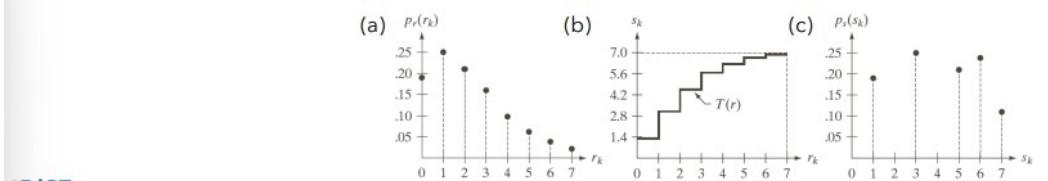
Es. di equalizzazione di una variabile casuale discreta

Supponiamo

- Immagine di  $64 \times 64$  (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	

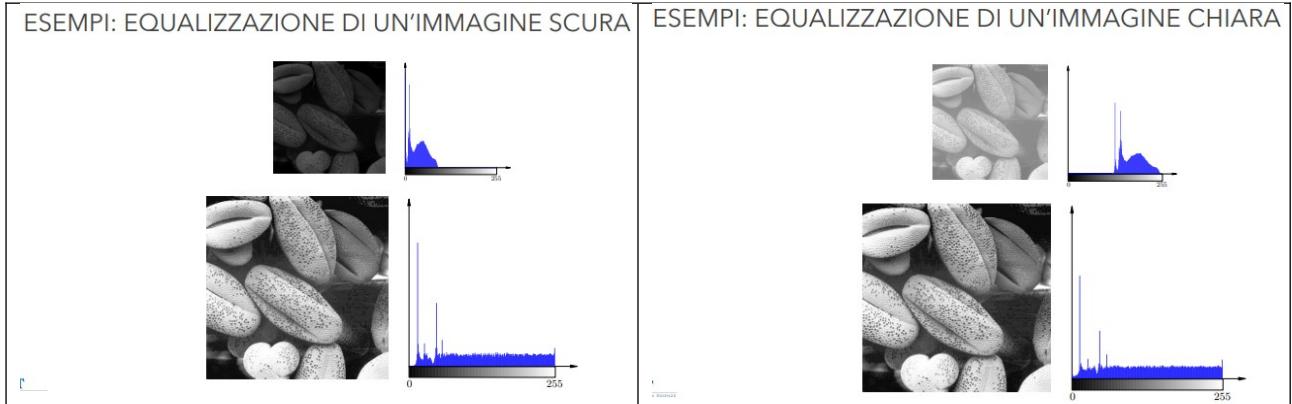
$$s_k = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad k = 0, \dots, 7$$



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

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

Di seguito abbiamo esempi di equalizzazioni di immagini chiare e scure, ma il risultato sarebbe uguale anche con immagini a basso e ad alto contrasto.



E' inoltre utile avere la possibilità di **ottenere un'immagine con una specifica (dell'istogramma) distribuzione (matching dell'istogramma)**, ed in tal caso bisogna trovare la **trasformazione F tale che  $z = F(r)$ , a partire da un'immagine di input** in cui i pixel sono distribuiti con una densità di probabilità  $p_r$  e dalla distribuzione di intensità desiderata  $p_z$ .

Considerando 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 con  $G(z) = s = T(r)$ .

In questo caso, il mapping cercato F, tale che  $z = F(r)$ , può essere ottenuto come

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

**Lavorando con gli istogrammi, che sono funzioni discrete, gli integrali diventano sommatorie**, come già discusso, e quindi

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

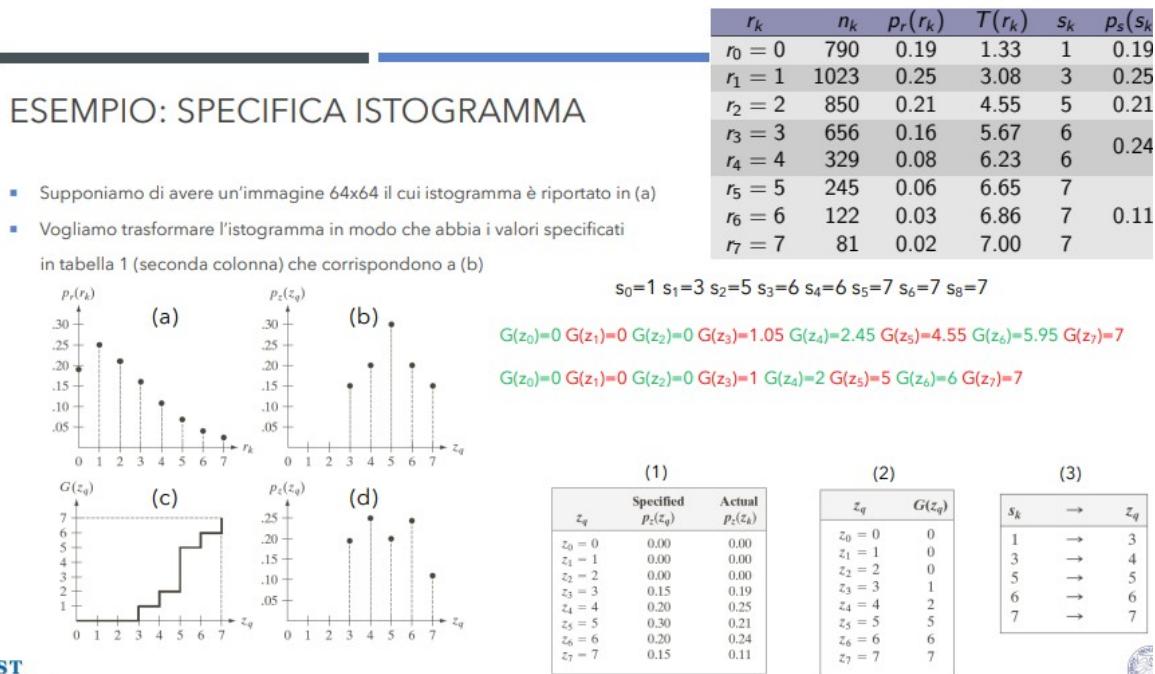
$$z_q = G^{-1}(s_k)$$

La **procedura di matching** dell'istogramma può quindi essere realizzata seguendo **alcuni passi**:

1. **ottenere  $p_r$  (densità di probabilità dei pixel) dall'immagine di input;**

2. **ottenere il mapping T utilizzando la relazione di equalizzazione soprastante per calcolare  $s_k$ , arrotondando il valore di quest'ultimo a numeri compresi in  $[0, L-1]$ ;**
3. **ottenere il mapping G dalla  $p_z$  (densità di probabilità dei pixel) specificata nella relazione soprastante per calcolare  $G(z_q)$ , arrotondando il valore di quest'ultimo a numeri compresi in  $[0, L-1]$ ;**
4. per ogni  $s_k$  (intensità nell'immagine di output) vengono utilizzati i valori di G per trovare i  $z_q$  corrispondenti, in modo che  $G(z_q)$  sia il più vicino a  $s_k$ . Nel caso in cui ci siano più valori di  $z_q$  per cui  $G(z_q)$  è il più vicino a  $s_k$ , bisogna scegliere il più piccolo;
5. **applicare la trasformazione F all'immagine originale tramite il mapping trovato al passo 4.**

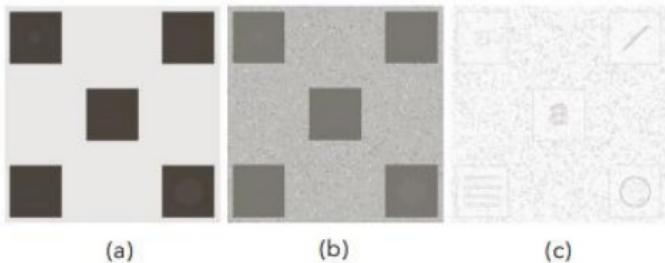
Perchè sia invertibile, G deve essere strettamente monotona, ma nei casi pratici tale proprietà viene soddisfatta raramente.



## Elaborazione locale dell'istogramma

Visto che l'equalizzazione dell'istogramma è un approccio globale, esiste anche un'equalizzazione locale. Questa è realizzata selezionando, per ogni pixel, un'area di vicinanza adeguata, su cui viene calcolata l'equalizzazione dell'istogramma (o matching). Questa procedura risulta essere più pesante computazionalmente, ma i pixel vicini condividono il più delle loro aree di vicinanza.

Esempio:



(a) immagine originale  
(b) immagine equalizzata  
(c) immagine equalizzata localmente (vicinanza 3x3)

## Statistiche da istogramma

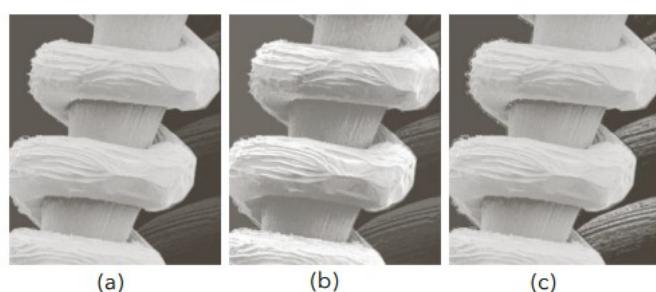
Possono essere facilmente elaborati alcuni **indici statistici dall'istogramma**:

- 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 inoltre 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



(a) immagine originale  
(b) immagine equalizzata  
(c) immagine migliorata con statistiche locali (vicinanza 3x3)

## ESEMPIO

$m_G$ : media intensità di tutti i pixel dell'immagine  
 $\sigma_G$ : deviazione standard intensità di tutti i pixel dell'immagine

- 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

$$\sigma_{S_{xy}} \leq k_2 \sigma_G$$

N.B.:

- per migliorare le zone chiare  
**k>1**
- Per migliorare le zone scure  
**k<1**

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

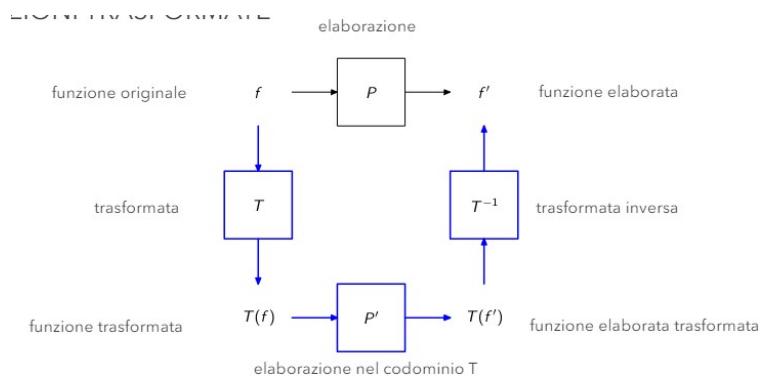
$$E = 4, k_0 = 0.4, k_1 = 0.02, k_2 = 0.4.$$

## Trasformata di Fourier (Lez.9)

### Funzioni trasformate

Tavolta può risultare più semplice **lavorare su una classe di funzioni se queste sono espresse in un altro dominio**, come può essere ad esempio **quello delle frequenze**, rispetto al solito dominio spaziale (o anche del tempo).

In questo modo una funzione  $f$  può sia essere direttamente elaborata ( $f'$ ) che sottoposta ad un'**operazione di trasformazione** ( $T(f)$ ), a cui poi segue l'elaborazione sulla **funzione trasformata** ( $T(f')$ ). In questo secondo caso bisognerà effettuare la **trasformazione inversa** ( $T^{-1}$ ) per ottenere la funzione di partenza ma elaborata ( $f'$ ).



## Fourier

Di queste trasformate noi considereremo solo quella di Fourier, alla cui base vi è l'idea che qualsiasi **funzione periodica può essere espressa come somma di**

**funzioni seno e coseno di frequenze diverse, ciascuna moltiplicata per un apposito coefficiente** (peso). Tale sommatoria prende il nome di **serie di Fourier**, che permette, dati sufficienti termini, di **esprimere la funzione periodica in modo esatto**.

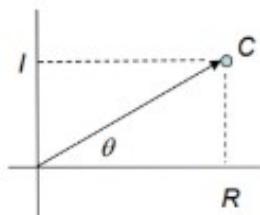
Fourier dimostrò che **anche funzioni non periodiche, a patto che l'area sotto la curva sia finita** (in modo che la trasformata converga, garantendo che la rappresentazione della funzione sia definita e significativa), **possono essere espresse come somma di seni e coseni moltiplicati per una funzione peso**. In tal caso non si parla più di serie di Fourier 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  $C = R + jI$ , con **R** che **denota la parte reale** e **I** (che sarebbe **i** e non **l**) che **denota la parte immaginaria**, ed ovviamente il suo **coniugato** (uguale al numero complesso ma con la parte immaginaria cambiata di segno)  **$C^*$**  è definito come  $C^* = R - jI$ .

I numeri complessi possono essere visualizzati geometricamente come punti in un piano in cui l'asse delle ascisse è l'asse reale (valori di **R**) e l'asse delle ordinate è l'asse immaginario (valori di **I**). Può essere utile, inoltre, **rappresentare tali numeri in coordinate polari**, quindi  $C = C(\cos \theta + j \sin \theta)$  o anche  $C = |C|e^{j\theta}$ , dove  $e^{j\theta}$  si ricava dalla **formula di Eulero**, per cui  $e^{j\theta} = \cos \theta + j \sin \theta$ . In tutto ciò, **theta** rappresenta l'angolo del vettore, quindi  $\theta = \arctan\left(\frac{I}{R}\right)$ , mentre 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 dimostrato da Fourier, come combinazione lineare di seni e coseni, moltiplicati per coefficienti opportuni:

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

con i che sarebbe j, quindi un numero immaginario, n che è l'indice delle frequenze armoniche, e 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 ( $c_n$ ). Come si può ben notare, la base della serie è composta da un insieme infinito di seni e coseni, in quanto maggiore è il numero di coefficienti, migliore sarà l'approssimazione.

## Impulso

La distribuzione delta di Dirac,  $\delta$ , o **impulso**, è definita come

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

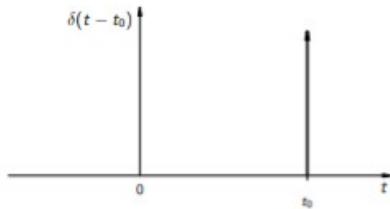
e

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

Questo è valido se l'impulso è **centrato nell'origine**, ma è possibile anche posizionarlo in un punto  $t_0$  arbitrario, ed 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**, che fornisce semplicemente, come risultato, il valore della funzione  $f(t)$  nella posizione dell'impulso  $t_0$ , con  $f(t)$  continua nella posizione dell'impulso.

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

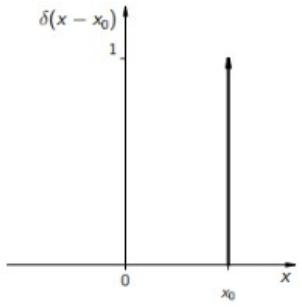
$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0)$$

## Impulso discreto

Per lavorare nel **discreto**, quindi con **funzioni definite solo in un insieme discreto di punti**, per cui  $f: Z \rightarrow R$ , l'**impulso discreto** può essere definito come

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

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



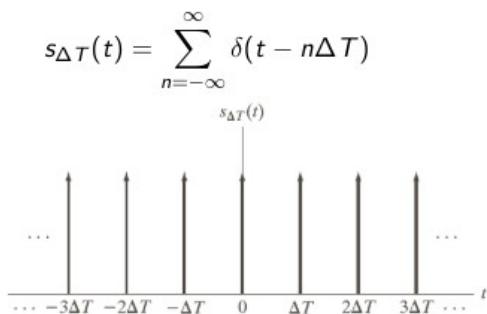
e anche in questo caso è soddisfatta la proprietà di sifting, stavolta rispetto alla sommatoria

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

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

## Treno di impulsi

Esiste poi anche il cosiddetto treno di impulsi  $s_{\Delta T}$ , definito come la **somma di infiniti impulsi periodici, centrati in  $n\Delta T$** , unitari e separati, con  $s_{\Delta T}$  che è quindi una funzione periodica con **periodo  $\Delta T$** .



## Trasformata di Fourier continua

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

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

e poichè  $t$  è integrata, come possiamo vedere, la **trasformata di Fourier di  $f(t)$**  è una funzione della variabile  $\nu$  (ni). Quindi, solitamente, la trasformata è indicata come

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

La trasformata di Fourier, come sappiamo, descrive  $f(t)$  come una combinazione lineare complessa di seni e coseni e, utilizzando la formula di Eulero

$$e^{i\theta} = \cos \theta + i \sin \theta,$$

possiamo esprimere l'equazione come

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

Data  $F(\nu)$  è possibile risalire a  $f(t)$  attraverso la **trasformata di Fourier inversa**, o **antitrasformata**.

$$f(t) = \mathcal{F}^{-1}\{F(\nu)\} = \int_{-\infty}^{\infty} F(\nu) e^{i2\pi\nu t} d\nu$$

Possiamo notare come **con la trasformata di Fourier si passi dal dominio del tempo a quello delle frequenze**, integrando rispetto a  $t$ , mentre **con l'antitrasformata passiamo dal dominio delle frequenze a quello del tempo**, integrando rispetto a  $\nu$ .

L'insieme della trasformata di Fourier e della sua antitrasformata prende il nome di **coppia di trasformate di Fourier**.

### Esempio di trasformata della funzione box

La funzione box, così chiamata poichè ha la **forma di una scatola**, è definita come

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

**Calcoliamo ora la trasformata di Fourier** a partire da questa funzione, ricordando che

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

e di conseguenza, nel nostro caso,

$$F(\nu) = \int_{-W/2}^{W/2} A e^{-i2\pi\nu t} dt$$

Dopo i vari passaggi di integrazione, utilizzando l'**identità trigonometrica** per cui

$$\sin \theta = \frac{(e^{j\theta} - e^{-j\theta})}{2j},$$

avremo

$$AW \frac{\sin(\pi\nu W)}{\pi\nu W},$$

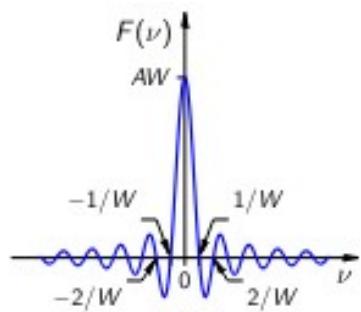
che equivale ad una **funzione** chiamata **sinc**

$$AW \text{sinc}(\nu W),$$

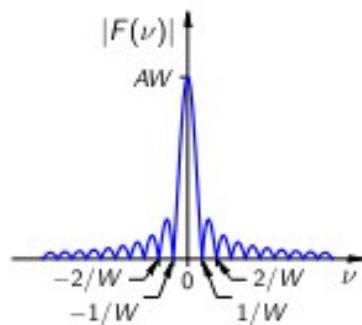
dove

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

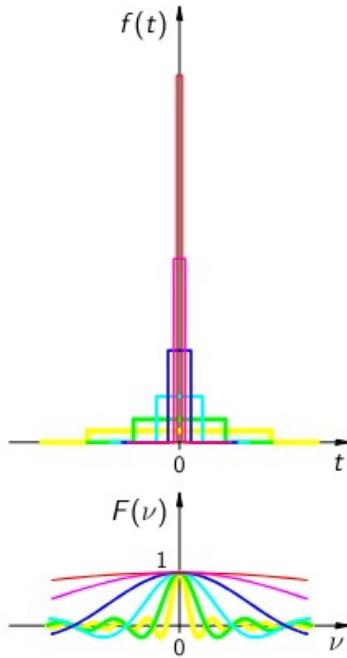
Il grafico della trasformata è



tuttavia, per agevolare la visualizzazione, si preferisce lavorare con la **magnitudo della trasformata**, detta anche **spettro di Fourier** (o spettro della frequenza).



Tra la **funzione box** e la sua **trasformata** si può osservare come ci sia una **proporzionalità inversa**, ossia più allarghiamo la box, quindi maggiore sarà il valore di  $W$ , più si restringe la funzione sinc, e viceversa.



Per questo motivo, la funzione box può essere anche sfruttata per calcolare la trasformata di Fourier di un impulso  $\delta$ . Infatti quando si considerano solo funzioni box dove  $AW = 1$  (A sarebbe altezza e W larghezza), facendo il limite per  $W \rightarrow 0$ , le funzioni box diventano sempre più strette e alte (quindi A tende all'infinito), mantenendo l'area sotto la curva costante a 1 (quindi l'integrale è 1, perché abbiamo impostato  $AW = 1$ ). In pratica, quando la larghezza della funzione box tende a 0 ( $W \rightarrow 0$ ), la sua trasformata di Fourier converge ad una funzione costante, che può essere interpretata come l'impulso.

Esempio di trasformata di un impulso e di un treno di impulsi

Utilizzando la proprietà di sifting dell'impulso, la sua trasformata di Fourier può essere calcolata come

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

con t che diventa 0 proprio per la proprietà di sifting.

Può inoltre essere mostrato che

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

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

## Convoluzione e trasformata di Fourier

La convoluzione tra due funzioni consiste nel **progressivo scorrimento di una delle due sull'altra**, combinando le informazioni delle due funzioni in ciascun punto.

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 **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, ossia la **trasformata di Fourier del prodotto** di due funzioni continue **consiste nella convoluzione delle trasformate** di  $f$  ed  $h$ .

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

Tale relazione prende il nome di **teorema di convoluzione**.

## Campionamento e trasformata di Fourier

Il campionamento è un'**operazione che permette di convertire una funzione continua in una sequenza di valori discreti**, in modo che quest'ultima possa essere elaborata (la sequenza) da un computer.

La **funzione campionata** è definita come il **prodotto tra 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.

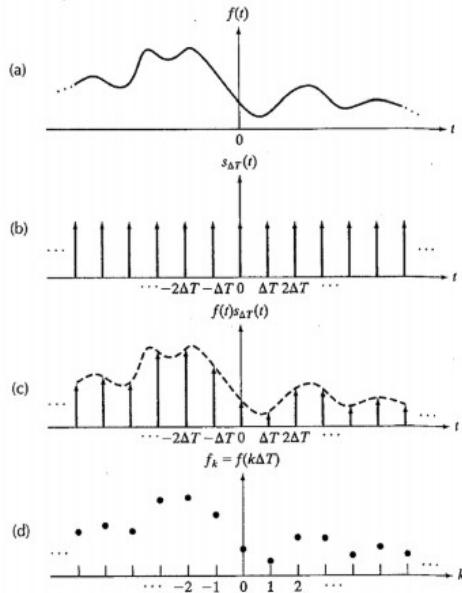
$$\tilde{f}(t) = f(t) s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t) \delta(t - n\Delta T)$$

Il **valore di ogni campione** è quindi determinato dalla “forza” dell'impulso pesato ottenuto dall'integrazione con  $f(t)$  (per la proprietà di sifting).

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

**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.

**Utilizzando il teorema di convoluzione, la trasformata di Fourier di una funzione campionata può essere calcolata come**

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

e dalla proprietà di sifting dell'impulso

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

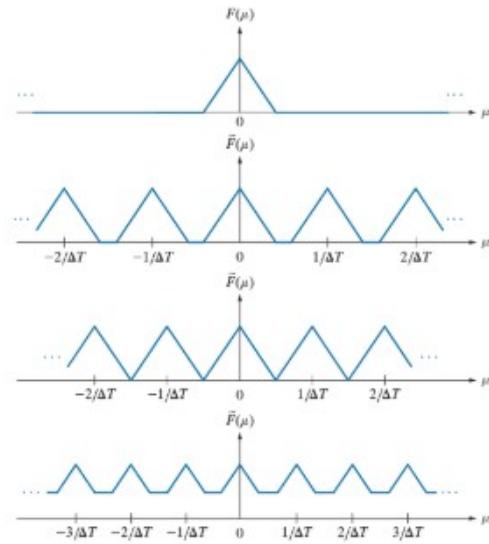
La trasformata di Fourier della funzione campionata è, quindi, una **sequenza periodica infinita di copie di  $F(\nu)$** , ossia la trasformata di Fourier della funzione originale  $f(t)$ , con periodo  $1/\Delta T$ .

La **trasformata della funzione campionata**, a differenza della funzione campionata stessa, è **una funzione continua**, poiché costituita da copie di  $F(\nu)$ , che a sua volta è una funzione continua.

La **separazione tra le copie** è regolata da  $1/\Delta T$ , che rappresenta la **frequenza di campionamento** utilizzata per generare le funzioni campionate. Possiamo distinguere **vari risultati di campionamento**:

- **sovracampionamento**, quando la frequenza è abbastanza alta da permettere un sufficiente distacco tra i vari periodi, riuscendo a preservare l'integrità di  $F(\nu)$ ;

- **campionamento critico** (o **tasso di Nyquist**), quando la frequenza di campionamento è alta appena da permettere di preservare l'integrità di  $F(v)$ ;
- **sottocampionamento**, quando la frequenza di campionamento è più bassa del minimo richiesto, non riuscendo a mantenere distinte le copie di  $F(v)$  e, di conseguenza, non riuscendo a preservarne l'integrità (di  $F(v)$ ).



## 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**.

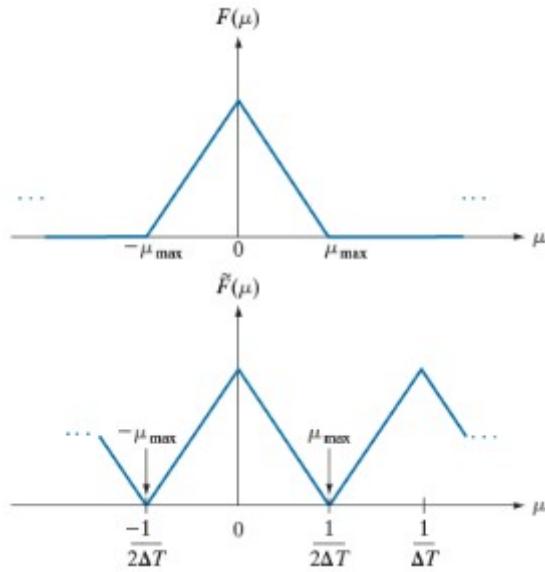
Possiamo ricostruire  $f(t)$ , a partire dalla sua **versione campionata**, se **isoliamo una copia di  $F(v)$  dalla sequenza periodica di sue copie** contenute nella **trasformata della funzione campionata**. Ovviamenete, una volta ottenuta la copia,  $f(t)$  sarà data dall'antitrasformata di  $F(v)$ .

E' possibile estrarre una copia di  $F(v)$  dalla sequenza periodica solo 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

$$1/\Delta T > 2v_{max}.$$

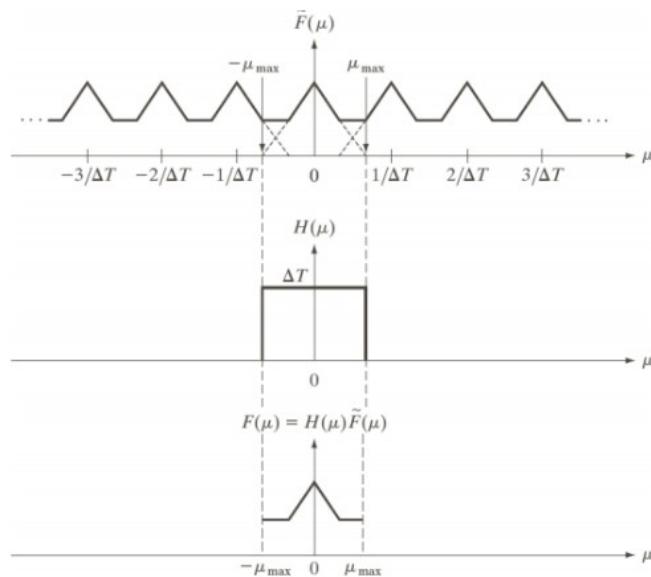
La relazione appena esplicitata equivale al cosiddetto **teorema di campionamento**.

Un tasso di campionamento esattamente uguale al doppio della frequenza più alta è il **tasso di Nyquist**, ossia il **tasso di campionamento critico**.



## Aliasing

L'aliasing è un 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. Se la funzione non è campionata con il tasso di Nyquist, ossia quando  $\frac{1}{\Delta T} < 2v_{max}$ , si verifica un **sottocampionamento** che porta le componenti tra  $\frac{1}{2\Delta T}$  e  $v_{max}$  a miscelarsi con altre componenti, a causa della sovrapposizione dei periodi. Tale effetto prende, appunto, il nome di aliasing.

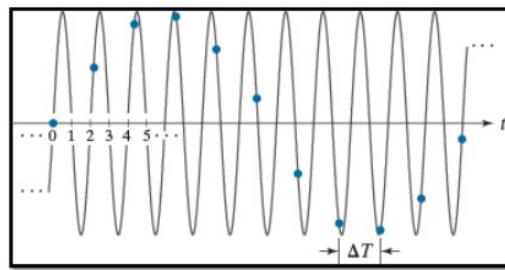


Nei casi pratici, l'aliasing risulta quasi inevitabile, sebbene gli effetti possano essere ridotti attenuando le componenti ad elevata frequenza (cioè effettuando smoothing) prima di eseguire il campionamento.

## ALIASING: ESEMPIO

Funzione originale:  $\sin(\pi t)$

Periodo:  $P=2s$



- La funzione campionata appare come una funzione sinusoidale che ha un frequenza molto più piccola della frequenza della funzione originale
- Il periodo di  $\sin(\pi t)$  è 2s, quindi gli zeri occorrono ogni secondo.  $\Delta T$  è la separazione tra i periodi

## Ricostruzione di una funzione campionata

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

Ricordando che la trasformata della funzione campionata è

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

moltiplichiamola per una funzione box  $H$ , in modo da **ottenere un periodo della trasformata della funzione originale**, ossia  $F(\nu)$ ,

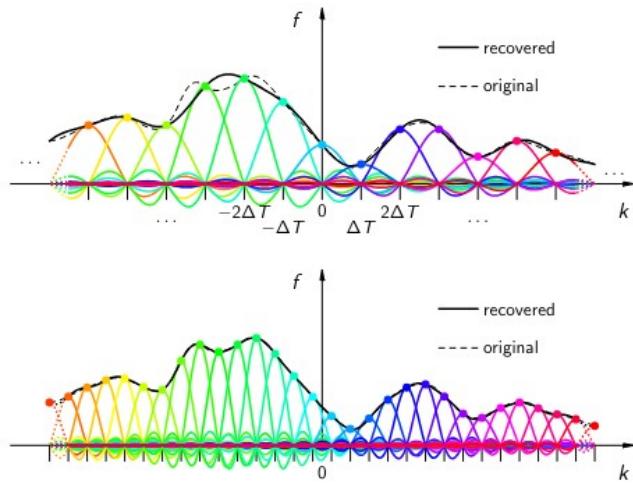
$$F(\nu) = H(\nu) \tilde{F}(\nu)$$

A questo punto, può essere applicata l'**antitrasformata per ottenere la funzione  $f(t)$  di partenza**.

Può essere inoltre mostrato che

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

visto che  $f(t)$  è uguale a  $f_k$  con  $t = k \Delta T$ .



## Trasformata di Fourier discreta (Lez.10)

Come si poteva esprimere la trasformata di una funzione a partire dalla funzione stessa, anche per la trasformata della funzione campionata

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

possiamo fare la stessa cosa, quindi esprimerla in termini della funzione campionata

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

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

Da questo è poi possibile dimostrare che

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

Poichè la **trasformata della funzione campionata è periodica**, tutta l'informazione della stessa è contenuta in un singolo periodo, e **alla base della trasformata di Fourier discreta vi è proprio il campionamento di un periodo della trasformata della funzione campionata.**

Se consideriamo **M campioni della trasformata della funzione campionata in un periodo,  $1/\Delta T$** , le frequenze considerate sono

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

che sostituite alle frequenze  $\nu$  in

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

permettono di calcolare il **valore della trasformata negli M campioni**, che prende il nome di **trasformata di Fourier discreta**

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

Gli M campioni possono poi essere ricostruiti mediante la **trasformata di Fourier discreta inversa**

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

La coppia di trasformata di Fourier discreta e della sua antitrasformata (che sono effettivamente delle sequenze), rappresentate solitamente come

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

quindi in funzione di u e di x, invece che di m ed n, prende il nome di **coppia di trasformate di Fourier discrete**.

Può essere inoltre dimostrato che **le due funzioni soprastanti sono periodiche**, nonostante la funzione originale non lo sia.

$$F(u) = F(u + kM) \quad \text{and} \quad f(x) = f(x + kM), \quad k \in \mathbb{Z}$$

## Convoluzione circolare

La convoluzione di una sequenza finita di M elementi può essere definita come

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

ed è definita **circolare in quanto la funzione risultante è periodica**, con la periodicità che deriva delle due funzioni f ed h.

Attraverso questa operazione, il [teorema di convoluzione](#) per la trasformata di Fourier continua può essere esteso anche a quella discreta.

## Relazione tra intervalli di campionamento e di frequenza

Se  $f(x)$  consiste di  $M$  campioni di una funzione  $f(t)$ , spaziati tra loro di  $\Delta T$ , la sequenza  $(f(x))$  copre un intervallo  $T = M \Delta T$ .

Nel dominio delle frequenze (quindi nel discreto), la spaziatura dei campioni  $\Delta u$  segue da

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

ed è data da

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

La trasformata di Fourier discreta è quindi definita su un range di frequenze, attraversato dagli  $M$  campioni, pari a

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

Ne consegue che, nel dominio delle frequenze, la spaziatura dei campioni dipende dalla lunghezza dell'intervallo campionato nel dominio spaziale, ossia  $T$ , mentre il range delle frequenze coperte dalla trasformazione di Fourier discreta dipende dal passo di campionamento  $\Delta T$ .

## Trasformata di Fourier per immagini (2D)

Tutti i concetti introdotti per il dominio monodimensionale possono essere estesi in modo abbastanza naturale anche al caso multidimensionale.

In particolare, siamo interessati al caso bidimensionale, visto che le immagini ne sono un esempio.

## Impulso 2D

La distribuzione delta di Dirac,  $\delta$ , o impulso 2D, è definita come

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

e

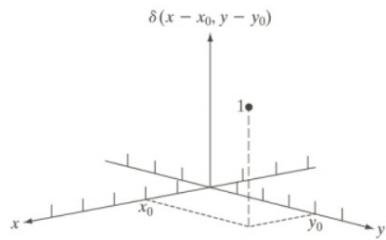
$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t, z) dt dz = 1$$

Anche in questo caso, ovviamente, è valida la proprietà di **sifting**, per cui l'integrale doppio di una funzione moltiplicata per l'impulso centrato in  $(t_0, z_0)$  equivale al valore della funzione nella posizione dell'impulso.

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

Come per l'impulso monodimensionale, anche per l'impulso bidimensionale vi è la **versione discreta**, per cui

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



### Coppia di trasformate di Fourier continue 2D

Nel caso bidimensionale, ovviamente, le trasformate sono funzioni di due variabili, quindi

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

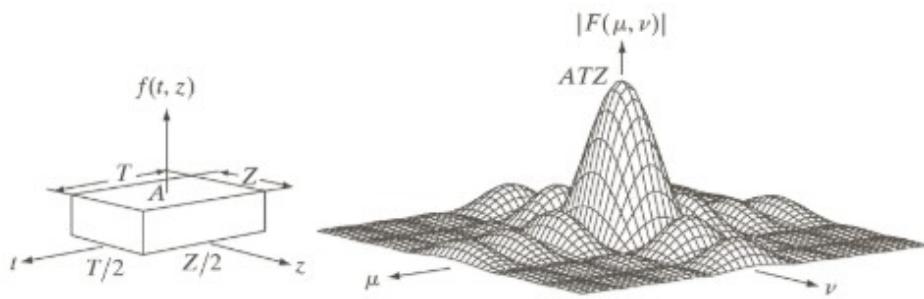
$$f(t, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\nu, \mu) e^{i2\pi(\nu t + \mu z)} d\nu d\mu$$

### Esempio di trasformata della funzione box 2D

Se volessimo calcolare la **trasformata della funzione box**, quindi, avremmo **due dimensioni**, T e Z:

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

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

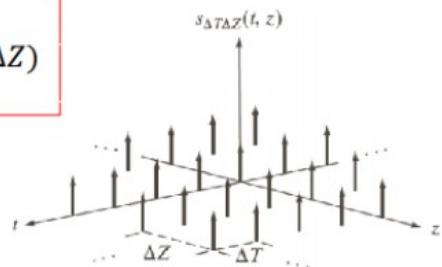


## Teorema di campionamento 2D

Come per il caso monodimensionale, **il campionamento può essere modellato a partire dal prodotto tra una funzione bidimensionale ed un treno di impulsi, anch'esso bidimensionale**. Il treno di impulsi non terrà più conto solo di  $\Delta T$  ma anche di  $\Delta Z$ , che equivalgono alla spaziatura che c'è tra i campioni lungo l'asse delle  $t$  e quella delle  $z$  della funzione.

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

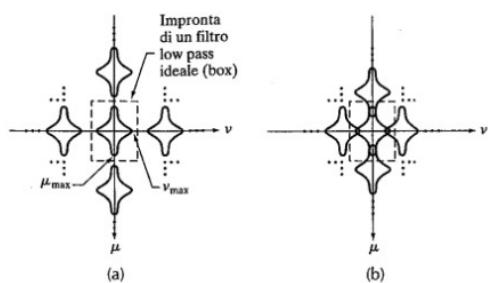
$$s_{\Delta T \Delta Z}(t, z) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(t - m\Delta T, z - n\Delta Z)$$



Anche in questo caso il **tasso di campionamento** deve essere almeno il doppio della frequenza massima in  $f(t, z)$ , ossia

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

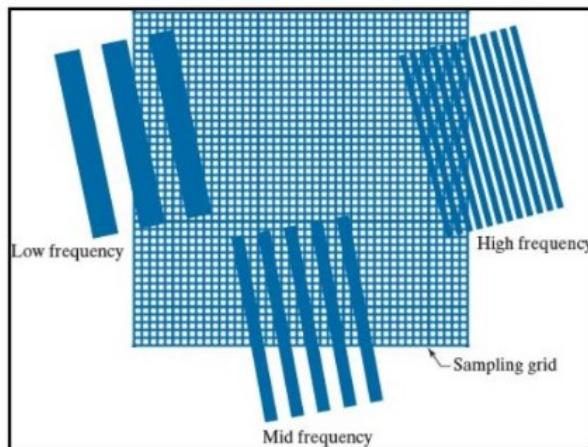
Inoltre se non si rispetta il tasso di Nyquist si genererà una situazione di **aliasing**.



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

## Aliasing nelle immagini digitali

Per spiegare gli effetti dell'aliasing **immaginiamo di avere una griglia che rappresenta un treno di impulsi, a cui sovrapponiamo un'immagine**. In base alla frequenza otterremo risultati diversi, ed in particolare, **maggior è la frequenza e più evidente sarà l'aliasing**. L'aliasing comporta la presenza di **frastagliamenti di linee o di veri e proprio artefatti**, ossia pattern che non appartengono all'immagine originale (come nell'immagine ad alta frequenza sottostante).



L'aliasing si genera anche quando si effettua un'operazione di ricampionamento, ossia di **resizing**, di un'immagine. Per ridurne gli effetti, in questa situazione, è possibile eseguire un **leggero smoothing** dell'immagine, prima di effettuare il ricampionamento.

(a) Immagine originale

(b) Immagine ricampionata

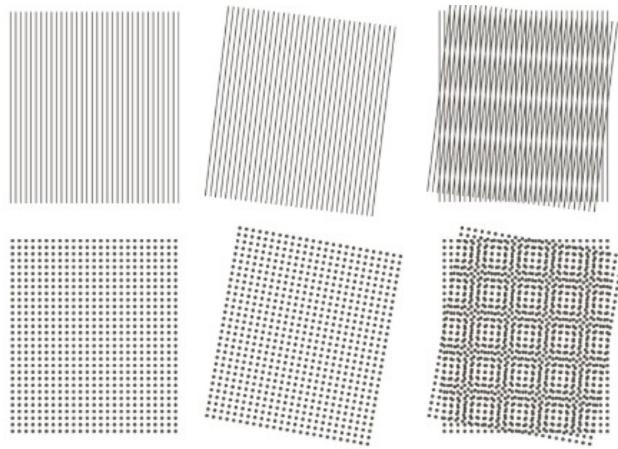
(c) applicazione dello smoothing prima del ricampionamento

Oss.: il ricampionamento è stato fatto attraverso la cancellazione di righe e colonne



## Effetto Moirè

L'effetto Moirè è un **artefatto causato dalla sovrapposizione di due pattern periodici**, come ad esempio due griglie. Un **esempio** di questo effetto si genera quando si fa una foto al monitor del PC, con i pattern dell'immagine che interferiscono con quelli dello schermo del computer.



## Coppia di trasformate di Fourier discrete 2D

Nella coppia di trasformate di Fourier discrete bidimensionale bisogna semplicemente aggiungere la seconda dimensione, e quindi

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

dove

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

$$\begin{aligned} u &= 0, \dots, M-1 & v &= 0, \dots, N-1 \\ x &= 0, \dots, M-1, & y &= 0, \dots, N-1 \end{aligned}$$

## Proprietà della trasformata di Fourier discreta 2D

La trasformata di Fourier discreta ha **diverse proprietà**:

- **traslazione**, in cui il prodotto tra la funzione originale ed un certo esponenziale trasla l'origine della trasformata di Fourier discreta e, viceversa, moltiplicando la trasformata per il negativo dello stesso esponentiale, viene traslata l'origine della funzione di partenza

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

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

- **rotazione**, in cui, rappresentando le variabili in **coordinate polari** e ruotando la funzione originale di un angolo  $\theta_0$ , anche la sua trasformata è ruotata dello stesso angolo, e viceversa;
- **periodicità**, infatti, come nel caso monodimensionale, le trasformate di Fourier discrete 2D sono periodiche, cioè

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

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

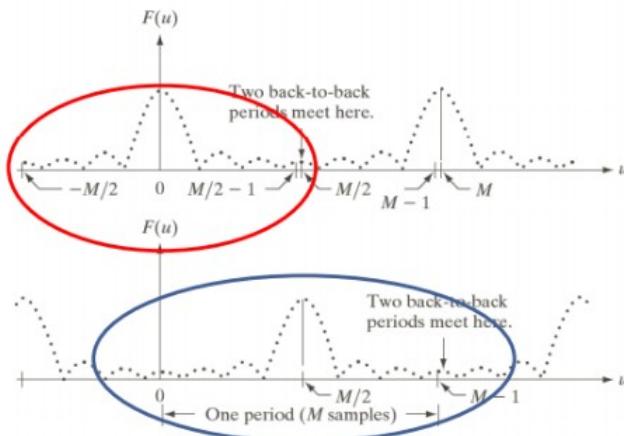
con  $k_1$  e  $k_2$  numeri interi.

I dati della trasformata nell'intervallo tra 0 e  $M-1$ , tuttavia, consistono in due mezzi periodi che si incontrano nel punto  $M/2$ . Di conseguenza, sia per la visualizzazione che per il filtraggio, è più conveniente avere in questo intervallo un periodo completo della trasformata. Per traslare la trasformata in modo che sia centrata in  $(M/2, N/2)$ , invece che in  $(0, 0)$ , basta porre  $u_0=M/2$  e  $v_0=N/2$  nella formula per calcolare la traslazione di  $f(x,y)$ , che diventerà

$$f(x, y) e^{i\pi(x+y)},$$

ma visto che  $e^{i\pi}=-1$  per la formula di Eulero, si avrà

$$f(x, y) (-1)^{(x+y)}$$



- **simmetria**, in cui due funzioni si dicono **pari**, o simmetriche, se

$$f(x, y) = f(-x, -y) \quad [ f(x, y) = f(M-x, N-y) ],$$

oppure **dispari**, o antisimmetriche, quando

$$f(x, y) = -f(-x, -y) \quad [ f(x, y) = -f(M-x, N-y) ]$$

**Le proprietà di simmetria della funzione originale corrispondono alle proprietà che si riscontrano nella trasformata**, le quali sono utili durante l'elaborazione. Ad esempio se la funzione originale è reale e pari, anche la sua trasformata sarà reale e pari.

	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

## Spettro di Fourier e angolo di fase

La trasformata di Fourier discreta 2D può essere espressa in forma polare, come

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

con la **magnitudo** (modulo) che è chiamata **spettro di Fourier**, e viene calcolata come

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

e  $\phi(u, v)$  che è l'**angolo di fase**, definito come

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

Lo **spettro di potenza**, invece, è definito come

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

La trasformata di Fourier discreta 2D, nella sua posizione (0,0), ha la **somma delle intensità dei pixel dell'immagine originale**, visto che

$$F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y),$$

sostituendo alla formula  $u = 0$  e  $v = 0$ .

Da questo, moltiplicando e dividendo per  $M \cdot N$ , ossia le dimensioni totali dell'immagine, risulta che

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

con  $\bar{f}$  che è il **valore medio** della funzione originale  $f$ .

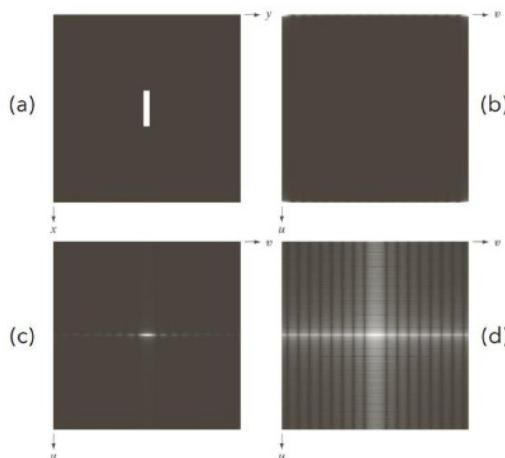
Generalmente, visto che la trasformata di Fourier discreta 2D, nella sua posizione (0,0), è molto più grande degli altri termini della trasformata, è necessaria una trasformazione logaritmica per visualizzarla.

**F(0,0)**, inoltre, è **detta componente dc**, ossia **direct current**, cioè componente che corrisponde alla frequenza zero.

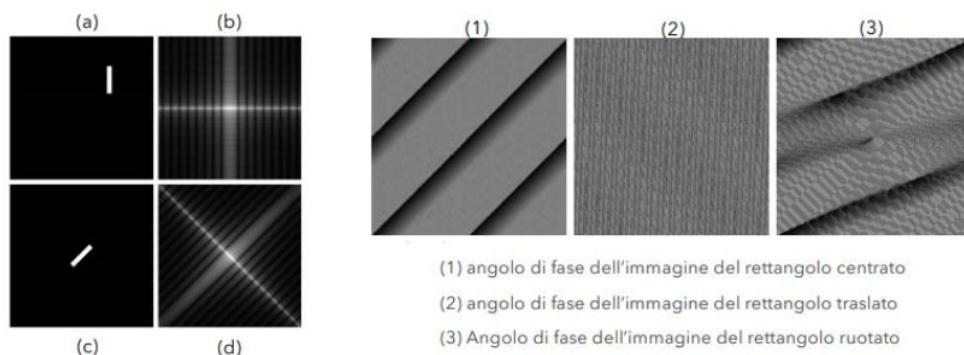
In generale, **lo spettro è una matrice le cui componenti determinano le intensità dell'immagine**, mentre **l'angolo di fase è una matrice che fornisce numerose informazioni sulla posizione degli oggetti nell'immagine**.

*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*



## Teorema di convoluzione 2D

Il teorema di convoluzione 2D risulta essere analogo a quello [1D](#), con la differenza che abbiamo due variabili.

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

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

## Convoluzione e errore di wraparound

Per calcolare la convoluzione, in questo caso, si deve utilizzare la convoluzione circolare, e quindi

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

Bisogna però tener conto della periodicità, infatti la convoluzione circolare di due funzioni periodiche può causare il cosiddetto **errore di wraparound**.

L'errore di wraparound si verifica quando, a causa della vicinanza dei periodi, questi (i periodi) interferiscono l'un l'altro.

Il problema può essere risolto utilizzando lo **zero padding**, ossia date due sequenze di campioni A e B, si aggiungono degli 0 in modo che entrambe abbiano P elementi

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

Nel caso delle immagini, supponendo di averne due (di immagini) di dimensioni Ax B e CxD, eseguiamo lo zero padding lungo righe e colonne, per ottenere

$$P = A+C-1$$

$$Q = B+D-1,$$

con immagini risultanti di dimensioni PxQ.

Se ambo le immagini hanno dimensioni uguali, ovviamente, P = 2M – 1 e Q = 2N – 1, anche se è preferibile che P = 2M e Q = 2N poichè gli algoritmi per il calcolo della trasformata di Fourier discreta sono più veloci quando lavorano con dimensioni pari.

Nel caso in cui una funzione non è 0 alla fine dell'intervallo, però, lo zero padding introduce [artefatti](#), ossia componenti ad alta frequenza nella trasformata. Per mitigare questo effetto si possono utilizzare delle tecniche di **windowing**,

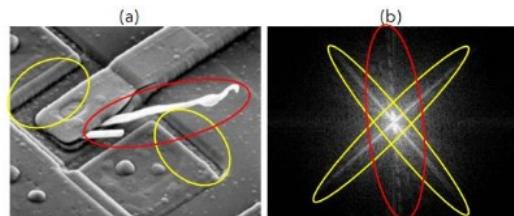
come ad esempio moltiplicare la funzione campionata per una Gaussiana, prima di effettuare il padding.

## Filtraggio nel dominio delle frequenze (Lez.11-12)

### DA ORA IN POI CI RIFERIREMO ALLA TRASFORMATA DI FOURIER DISCRETA SEMPLICEMENTE COME TRASFORMATA DI FOURIER

La trasformata di Fourier di un'immagine non fornisce informazioni tali da poter effettuare un'associazione diretta con le caratteristiche spaziali dell'immagine originale, ma si possono fare alcune **considerazioni a partire dalle sue componenti in frequenza**. Ad esempio, alcune **informazioni sulla struttura della scena possono essere dedotte dallo spettro**, come nell'esempio sottostante.

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 parallelismi rispetto all'immagine.



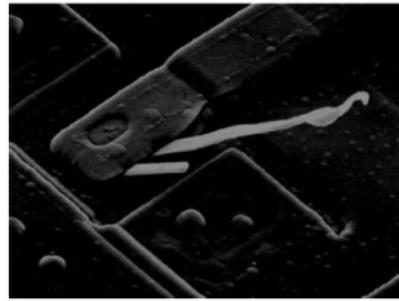
**Il filtraggio nel dominio delle frequenze viene ottenuto** modificando la trasformata di Fourier di un'immagine  $f(x,y)$  e poi applicando l'antitrasformata, in modo da ottenere l'immagine elaborata.

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

- $H(u,v)$  è la funzione filtro (o funzione di trasferimento del filtro)
- $g(x,y)$  è l'immagine filtrata
- $F, H$ , e  $g$  sono array della stessa dimensione  $M \times N$  dell'immagine originale
- Se  $H$  è reale e **simmetrica** e  $f$  è reale,  $g$  è reale
  - Le componenti immaginarie dovute ad errori numerici possono essere ignorate

**Un primo semplice filtro potrebbe essere creato a partire dalla componente dc della trasformata di Fourier**, ossia la componente di posizione  $(0,0)$ , che come sappiamo corrisponde all'intensità media della funzione originale  $f$ .

Possiamo quindi costruire un semplice filtro  $H(u,v)$  che avrà valore 0 quando  $u$  e  $v$  saranno entrambi uguali a 0 ( $u = v = 0$ ) e avrà valore 1 negli altri casi. Così facendo impostiamo a 0 solo la componente dc, il che ha l'effetto di spostare a 0 la media dell'immagine filtrata  $g$ , rendendola ovviamente più scura.



## Filtri passa basso e passa alto

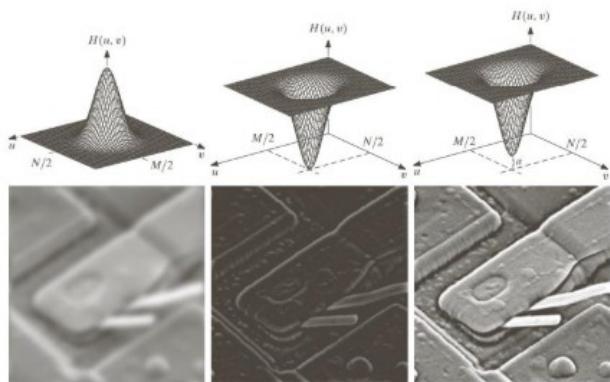
Il filtro visto in precedenza, in realtà, non è molto utile e lo utilizziamo solo come esempio. Procediamo ora alla spiegazione di alcuni filtri decisamente più utili.

**Il filtro passa basso è un filtro che attenua le alte frequenze, lasciando inalterate quelle basse** (frequenze). Il risultato di un'immagine filtrata tramite passa basso è una copia sfocata dell'immagine originale.

**Il filtro passa alto, invece, è un filtro che preserva le alte frequenze, attenuando quelle basse** (frequenze), ed in particolare la dc dovrebbe sparire. Il risultato di un'immagine filtrata tramite passa alto porta ad un miglioramento dei dettagli (sharpening), ma con contrasto minore. E' inoltre possibile aggiungere una piccola costante al filtro, in modo che le informazioni associate alla componente dc vengano 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.



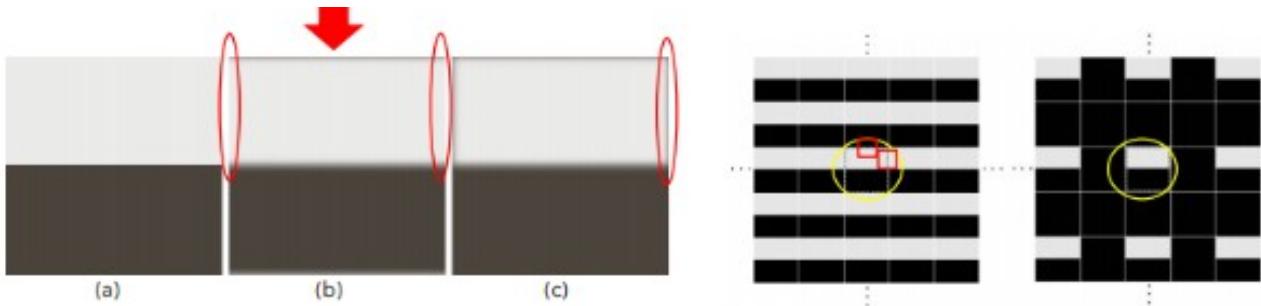
## Wraparound e padding

Ricordiamo la convoluzione circolare di due funzioni periodiche.

**Quando andiamo ad effettuare lo smoothing, ossia quando applichiamo un filtro passa basso, ad un'immagine senza padding, la sfocatura non sarà uniforme; caso contrario se invece aggiungiamo il padding.**

Consideriamo le seguenti immagini per spiegare questa situazione. In particolare abbiamo una prima immagine, che è quella originale, seguita da due versioni

smoothed, rispettivamente senza e con padding. Nella figura (b) notiamo come l'immagine non sia sfocata nei lati, a differenza della figura (c) che è sfocata in maniera uniforme. Il motivo è da ricercare nella periodicità della trasformata di Fourier discreta, la quale fa sì che, **senza padding, l'immagine originale venga ripetuta in ogni direzione nello stesso modo**. Ne consegue che nei lati chiari stiamo andando ad eseguire una sfocatura su pixel uguali, il che non porta ad alcun tipo di sfocatura. Con il padding, invece, l'immagine originale avrà in prossimità di tutti i bordi dei pixel scuri, e dunque **la sfocatura risulterà uniforme**.



Il padding applicato all'immagine originale, come sappiamo, può evitare errori di wraparound.

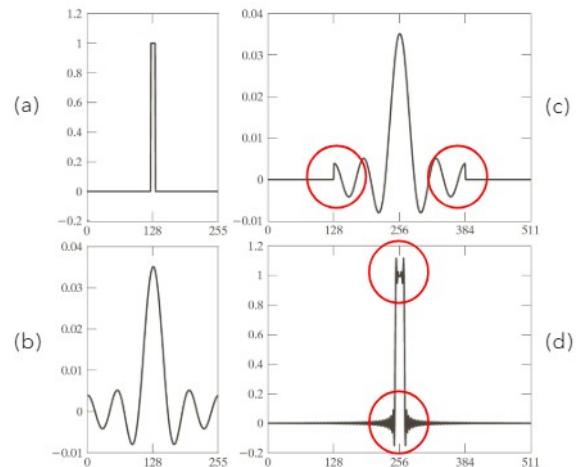
**Sia l'immagine che il filtro dovrebbero essere sottoposti a padding**, con questa operazione che dovrebbe avvenire **nel dominio spaziale**.

**Se il filtro è specificato nel dominio delle frequenze**, bisogna trasformarlo nel dominio spaziale, applicare il padding e poi ritrasformarlo nel dominio delle frequenze. Questa operazione può però comportare la comparsa del fenomeno del **ringing**, il quale causa errori considerevoli.

## Ringing

### RINGING

- (a) Il filtro ideale nel dominio delle frequenze
- (b) è trasformato nel dominio spaziale
- (c) padded
- (d) trasformato indietro nel dominio delle frequenze
  - L'effetto del ringing è evidente



Per spiegare il ringing, possiamo considerare la trasformata di Fourier di una funzione box, ovvero una funzione sinc con frequenze componenti che si

estendono all'infinito. Per questo motivo anche la trasformata inversa della funzione box deve avere componenti, stavolta spaziali, che si estendono all'infinito.

Di conseguenza **un qualsiasi troncamento spaziale del filtro per implementare lo zero padding introduce una discontinuità che comporta il fenomeno del ringing nel dominio della frequenza**.

In definitiva si può concludere che **non è possibile effettuare il padding della rappresentazione spaziale di un filtro definito nel dominio della frequenza per evitare gli errori di wraparound**.

Una **migliore procedura** consiste nel fare il padding dell'immagine e, successivamente, definire il filtro nel dominio della frequenza delle stesse dimensioni dell'immagine con padding. Ciò **provocherà comunque errori di wraparound**, visto che non è stato utilizzato un padding per il filtraggio, ma questi sono **mitigati dal padding dell'immagine** (ciò risulta essere comunque preferibile al ringing).

## Filtri di shift zero phase

Il filtri di shift zero phase, ossia **senza traslazione di fase**, sono tutti quei **filtri che si occupano di preservare l'angolo di fase**, in modo da non perdere informazioni sul contenuto dell'immagine stessa (che come possiamo vedere dalle immagini sottostanti avrebbe un impatto severo sull'output del filtraggio).



Questi filtri interessano, in egual misura, sia la parte reale che la parte immaginaria, di conseguenza il loro rapporto non cambia.

$$\begin{aligned} F(u, v) &= R(u, v) + iI(u, v) \\ g(x, y) &= \mathcal{F}^{-1}\{H(u, v)R(u, v) + iH(u, v)I(u, v)\} \end{aligned}$$



## Procedura per il filtraggio delle frequenze

1. Data un'immagine  $f(x,y)$ , di dimensioni  $M \times N$ , vengono calcolati i parametri di padding  $P$  e  $Q$ , che solitamente avranno valori  $P = 2M$  e  $Q = 2N$ ;
2. Si applica il padding all'immagine di partenza, ottenendo  $f_p$ ;
3. Opzionalmente possiamo moltiplicare l'immagine con padding  $f_p$  per  $(-1)^{x+y}$ , in modo da centrare la sua trasformata in  $(M/2, N/2)$ ;
4. Calcoliamo la trasformata di Fourier discreta dell'immagine con padding  $f_p$ ;

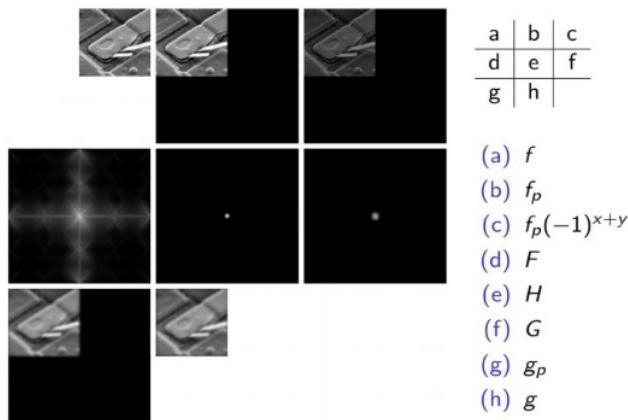
5. Si genera una funzione filtro simmetrica e reale  $H(u,v)$ , di dimensioni PxQ, optionalmente centrata in  $(P/2, Q/2)$ ;
6. Si modifica la trasformata di Fourier filtrandola, quindi  $G(u,v) = F(u,v) H(u,v)$ ;
7. Si ottiene l'immagine elaborata applicando l'antitrasformata alla funzione risultante dal precedente passo, prendendo solo la parte reale

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

8. Opzionalmente possiamo moltiplicare l'immagine elaborata per  $(-1)^{x+y}$  in modo da centrare l'immagine;
9. Il risultato finale  $g(x,y)$  viene ottenuto effettuando il cropping (ritaglio l'immagine, eliminando il padding aggiunto inizialmente) della regione MxN dell'immagine elaborata  $g_p$ .

*Esempio*

- |   |        |
|---|--------|
| a) immagine di partenza $f$ , di dimensioni MxN;  | .1     |
| b) estensione dell'immagine $f$ , $f_p$ , di dimensioni PxQ;                            | .2     |
| c) risultato della moltiplicazione di $f$ per $(-1)^{x+y}$ ;                            | .3     |
| d) spettro di $F_p$ ;   | .4 (?) |
| e) filtro $H$ di dimensioni PxQ;  | .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     |



## Progettazione di filtri spaziali

La conoscenza del dominio delle frequenze può essere utilizzata per guidare la progettazione dei filtri spaziali. Ad esempio può essere considerato il **filtro gaussiano** (monodimensionale in questo caso), che **nel dominio della frequenza** equivale a

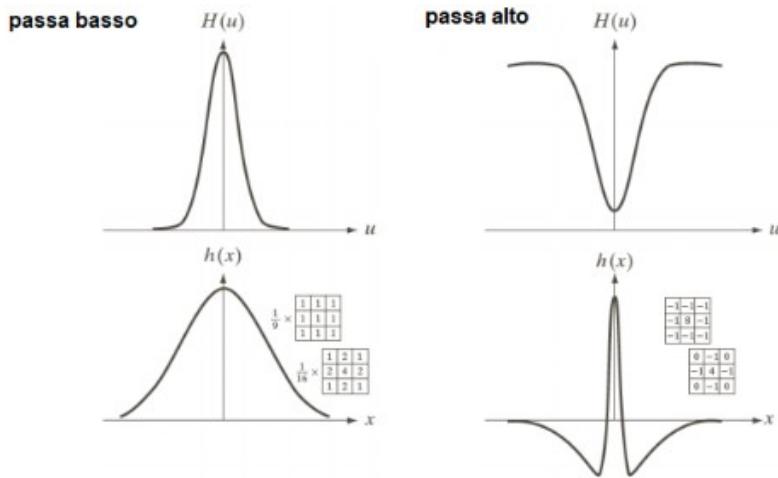
$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}},$$

con il **corrispondente filtro nel dominio spaziale** che viene ottenuto effettuando l'antitrasformata del filtro gaussiano nel dominio delle frequenze, ossia

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

Entrambe le trasformate appena viste, che formano una coppia, sono reali, quindi non abbiamo a che fare con numeri complessi, e sono gaussiane in ambo i domini.

Come si può facilmente immaginare, **tramite questi filtri si può effettuare un filtraggio di tipo passa basso**.



**Si possono progettare anche filtri più complessi**, come ad esempio un **filtro definito come differenza di due gaussiane**, il quale **corrisponde ad un filtro passa alto**

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

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

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

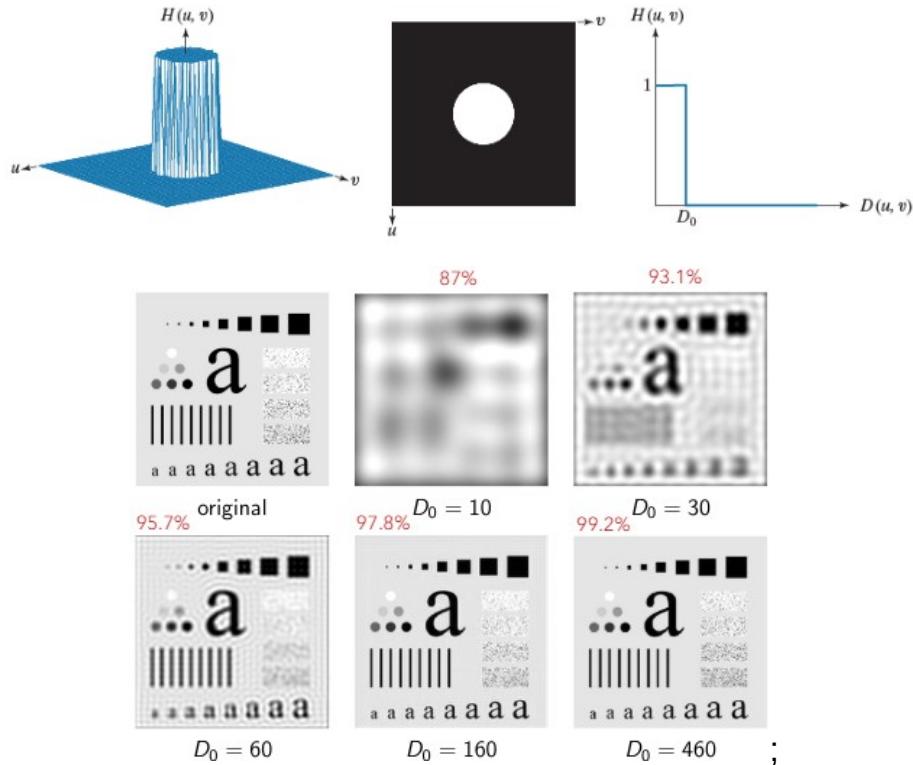
## Filtri passa basso

Esistono **diverse tipologie di filtri passa basso**:

- il **filtro passa basso ideale**, che è definito come

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

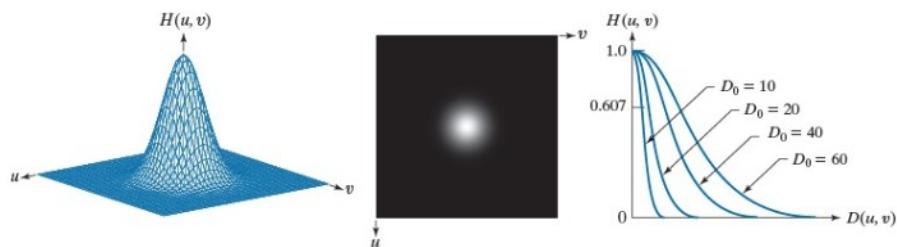
con  $D$  la **distanza tra i pixel  $(u,v)$  ed il centro del rettangolo delle frequenze** e  $D_0$  il **raggio del cerchio**, centrato nell'origine (ossia al centro del rettangolo delle frequenze, che sarebbe la rappresentazione delle frequenze dell'immagine nello spazio). In pratica se una frequenza si trova all'interno di questo cerchio di raggio  $D_0$  non viene attenuata, mentre nel caso in cui si trovi all'esterno, allora sì. Questo tipo di filtraggio, però, introduce un alto effetto di ringing.

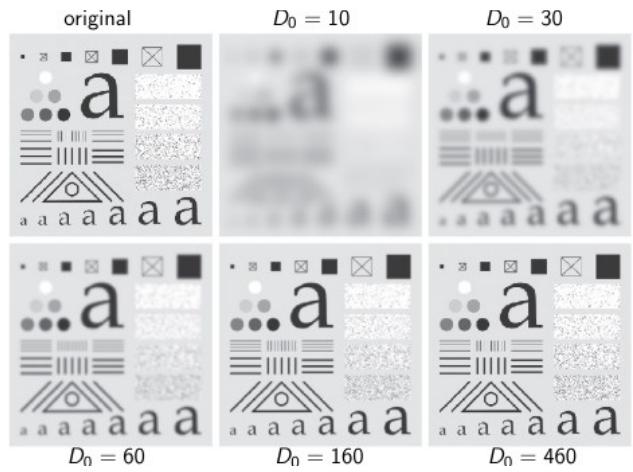


- il **filtro passa basso Gaussiano**, che è definito come

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

e fornisce all'immagine un effetto di **smoothing, non introducendo ringing**

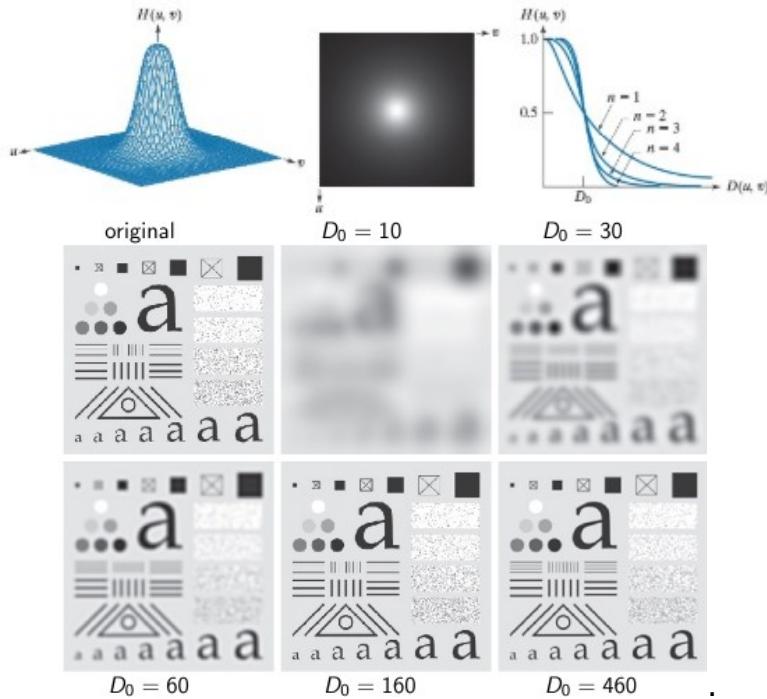




- il **filtro passa basso Butterworth**, che è definito come

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

e fornisce all'immagine un **effetto di smoothing maggiore rispetto a quello del filtro passa basso Gaussiano**, ma al contempo **introducendo il fenomeno del ringing**, che risulta essere comunque più accettabile rispetto a quello introdotto tramite filtro passa basso ideale. Bisogna però fare attenzione all'ordine n, poichè più grande è questo valore e maggiore sarà l'effetto del filtraggio, ma al contempo aumentano anche gli effetti del ringing.



**Nel filtro passa basso ideale il fenomeno del ringing è maggiormente presente a causa della marcata discontinuità delle frequenze**, che invece nel filtro Gaussiano e nel filtro Butterworth è graduale (la discontinuità delle frequenze).

## Filtri passa alto

I filtri passa alto si ottengono **sottraendo ad 1 il filtro passa basso corrispondente**, come segue

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

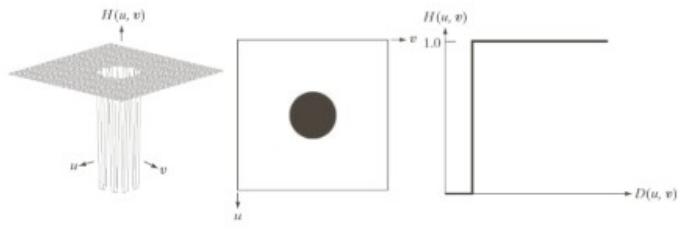
(HP sta per high pass, quindi passa alto, mentre LP sta per low pass, quindi passa basso).

Di conseguenza, come per i filtri passa basso, possiamo distinguere **3 tipi di filtri passa alto**:

- il **filtro passa alto ideale**, che è definito come

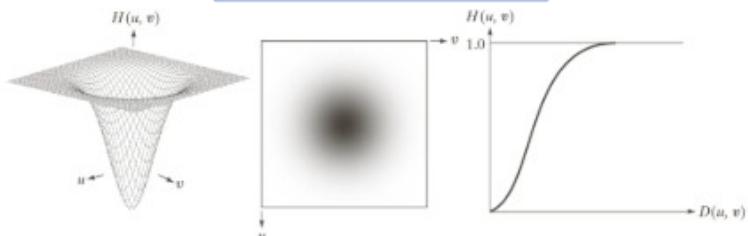
$$H(u, v) = \begin{cases} 0, & D(u, v) \leq D_0 \\ 1, & D(u, v) > D_0 \end{cases}$$

con tutte le frequenze che si trovano all'interno di un cerchio di raggio  $D_0$  centrale nell'origine che vengono attenuate, al contrario di ciò che accadeva nel filtro passa basso ideale



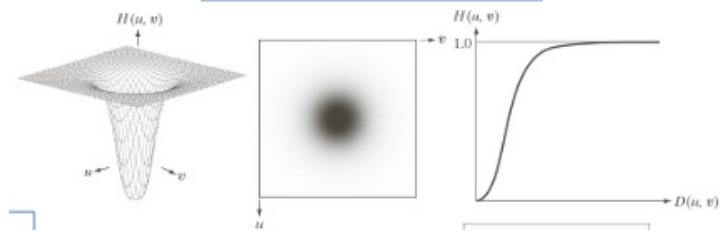
- il **filtro passa alto Gaussiano**, che è definito come

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

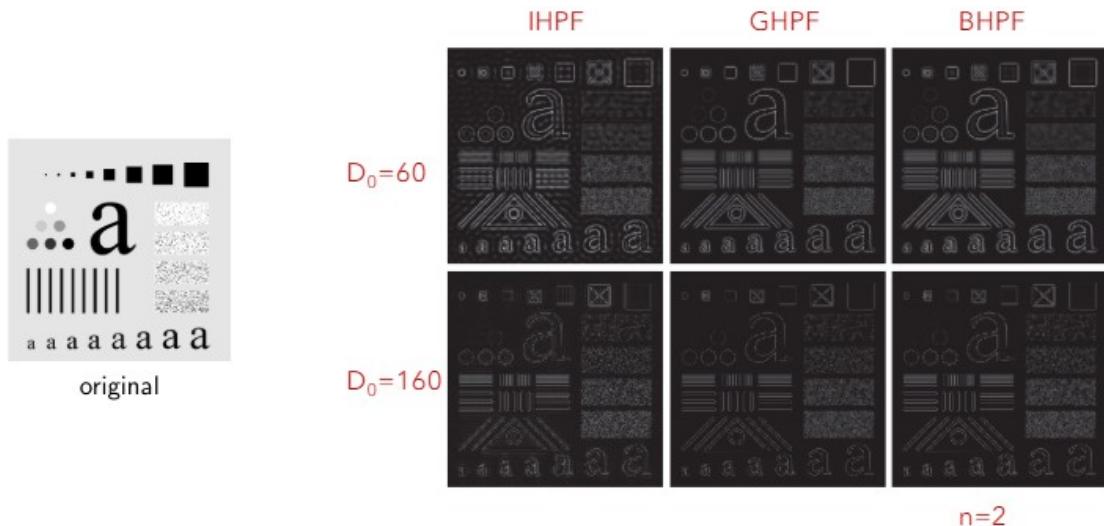


- il **filtro passa alto Butterworth**, che è definito come

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



Come si può notare dalle seguenti immagini, gli effetti dei tre filtri passa alto, in termini di ringing, smoothing, ecc., corrispondono a quelli che si avevano nel caso dei corrispondenti filtri passa basso.



## Laplaciano

Il **filtro Laplaciano** (che abbiamo visto nel dominio spaziale con Ferone) **nel dominio delle frequenze** è

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

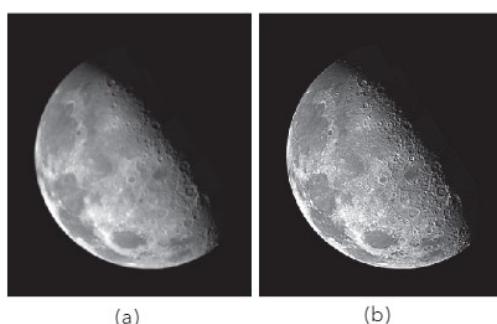
con l'**immagine Laplaciana** che viene ottenuta con l'antitrasformata

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

Infine, per ottenere l'**immagine filtrata**,

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

con  $c$  che è una costante che può assumere un valore compreso tra -1 e 0.



- (a) Immagine originale di blurry
- (b) Risultato del miglioramento Laplaciano nel dominio delle frequenze

## Unsharp masking

**Nel dominio delle frequenze è anche possibile creare un filtro per l'unsharp masking**, definito come

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

dove  $f_{\text{LP}}$  è un'**immagine sfocata**, analoga all'immagine originale  $f(x, y)$ , che si calcola a sua volta come

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

con  $H_{\text{LP}}$  che rappresenta un filtro passa basso.

**Nel dominio spaziale l'immagine filtrata** sarà ottenuta come

$$g(x, y) = f(x, y) + k g_{\text{mask}}(x, y) \quad \begin{cases} k = 1 \text{ unsharp masking} \\ k > 1 \text{ highboosting} \end{cases},$$

che con **k=1** permette l'**unsharp masking**, mentre con **k>1** definisce l'**highboosting** (migliora la nitidezza delle immagini in base al valore di k).

A partire dalle formule precedenti è possibile esprimere l'espressione dell'immagine filtrata nel dominio delle frequenze, come

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

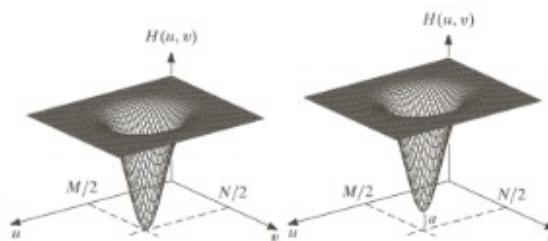
che può anche essere espresso in termini di filtro passa alto

$$g(x, y) = \mathcal{F}^{-1}\{[1 + k H_{\text{HP}}(u, v)] F(u, v)\}.$$

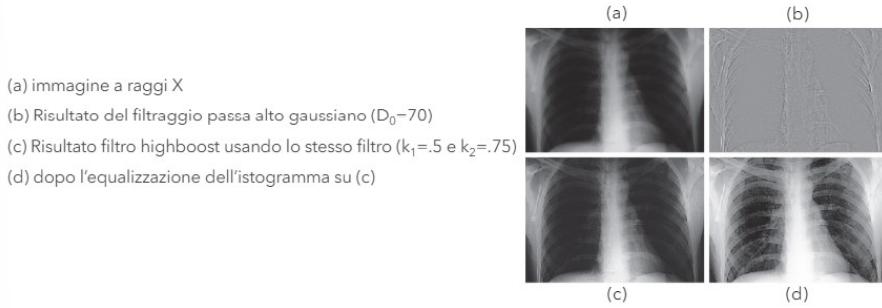
Generalizzando, quindi, l'immagine filtrata si ottiene come

$$g(x, y) = \mathcal{F}^{-1}\{(k_1 + k_2 H_{\text{HP}}(u, v)) F(u, v)\},$$

con  $k_1 \geq 0$  che è l'**offset dall'origine** e  $k_2 \geq 0$  che è il **contributo delle alte frequenze**.



## ESEMPIO DI MIGLIORAMENTO CON MASCHERAMENTO DI UNSHARP



## Filtraggio omomorfico

**L'aspetto di un'immagine può essere migliorato anche attraverso operazioni di compressione dei range dell'intensità e di miglioramento del contrasto, tramite la combinazione delle componenti di illuminazione (i) e riflettanza (r). Tale filtraggio prende il nome di omomorfico.**

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

Questa relazione non può essere espressa nel dominio delle frequenze poichè qui la trasformata di Fourier di un prodotto non è uguale al prodotto delle trasformate, a differenza del dominio spaziale.

$$\mathcal{F}\{f(x, y)\} \neq \mathcal{F}\{i(x, y)\} \mathcal{F}\{r(x, y)\}$$

Possiamo, tuttavia, esprimere l'immagine nella sua versione logaritmica, in modo da poter separare le due componenti nel dominio delle frequenze (a partire dalla proprietà dei logaritmi per cui il logaritmo di un prodotto equivale alla somma dei logaritmi).

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

A questo punto possiamo procedere all'applicazione della trasformata di Fourier discreta ( $Z(u, v)$ ) sulle singole componenti

$$Z(u, v) = \mathcal{F}\{z(x, y)\} = \mathcal{F}\{\log i(x, y)\} + \mathcal{F}\{\log r(x, y)\},$$

per poi applicare il filtro nel dominio delle frequenze (anche il filtro viene applicato ad entrambe le componenti)

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

( $F_i$  e  $F_r$  sono, rispettivamente, la trasformata dell'illuminazione e la trasformata della riflettanza).

Per tornare al dominio spaziale basterà invertire l'immagine logaritmica filtrata nel dominio delle frequenze  $S(u, v)$  utilizzando l'esponenziale.

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

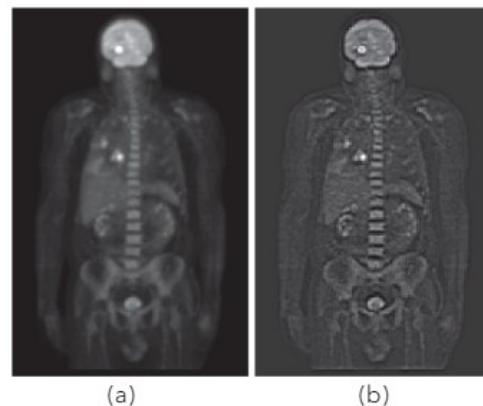
Questo processo può essere realizzato utilizzando un filtro come



$$H(u, v) = (\gamma_H - \gamma_L) \left( 1 - \exp \left( -c \frac{D^2(u, v)}{D_0^2} \right) \right) + \gamma_L$$

in cui, 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 dalle basse frequenze (illuminazione), amplificando al contempo quelli (gli effetti) dovuti alle alte frequenze (riflettanza).

### ESEMPIO DI FILTRO OMOMORFICO



- (a) Immagine scansione PET dell'intero corpo
- (b) miglioramento attraverso il filtraggio omomorfico con  $\gamma_L = 0.4$  e  $\gamma_H = 3$ ,  $D_0 = 20$

**Il vantaggio del filtraggio omomorfico** consiste nel migliorare l'immagine mantenendo il controllo sulle componenti di illuminazione e riflettanza.

### Filtraggio selettivo

Il filtraggio selettivo è un **filtraggio che permette di lavorare solo su specifiche bande di frequenza o piccole regioni del dominio di frequenza**.

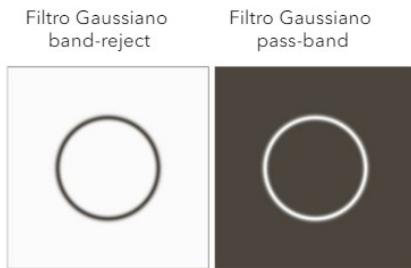
### Filtri band-reject e passa-banda

Della categoria del filtraggio selettivo che permette di **lavorare su specifiche bande** fanno parte i **filtri band-reject e passa-banda**, con i primi che si ottengono come

Ideal (IBRF)	Gaussian (GBRF)	Butterworth (BBRF)
$H(u,v) = \begin{cases} 0 & \text{if } C_0 - \frac{W}{2} \leq D(u,v) \leq C_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = 1 - e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}$	$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - C_0^2}\right]^{2n}}$

mentre i secondi si ottengono nella stessa maniera in cui otteniamo un filtro passa alto da un filtro passa basso, ossia

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$



## Filtri notch

I filtri notch **operano su regioni limitate del dominio di frequenza**, rigettando o lasciando passare frequenze specifiche.

Un esempio di questa categoria di filtri potrebbe essere un filtro progettato con valori 0 in corrispondenza di frequenze legate a rumore periodico, in modo da eliminarlo.

Poichè i **filtri zero-phase shift** (filtri che non introducono cambiamento nella fase delle frequenze del segnale quando viene filtrato) 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  $(-u_0, -v_0)$ .

Un **filtro notch di rigetto** è costruito come **prodotti di filtri passa alto**, in cui i centri sono traslati nei centri dei notch

$$H_{NR}(u, v) = \prod_{k=1}^Q H_k(u, v) H_{-k}(u, v)$$

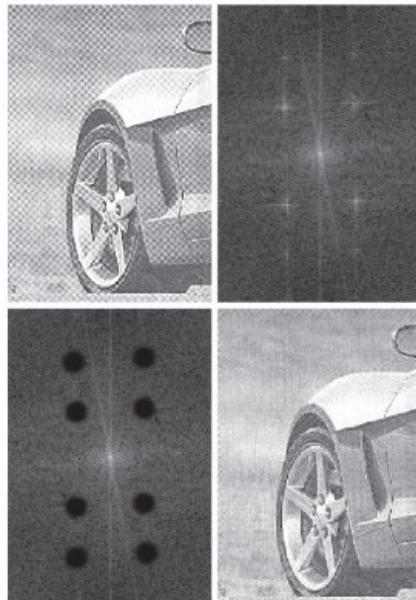
dove  $H_k(u, v)$  e  $H_{-k}(u, v)$  sono filtri passa alto con centri  $(u_k, v_k)$  e  $(-u_k, -v_k)$ , rispettivamente

I centri sono specificati rispetto al centro del rettangolo della frequenza ( $M/2$ ,  $N/2$ ) e la distanza per ogni filtro è calcolata come

$$D_k(u, v) = \left[ \left( u - \frac{M}{2} - u_k \right)^2 + \left( v - \frac{N}{2} - v_k \right)^2 \right]^{\frac{1}{2}} \text{ e } D_{-k}(u, v) = \left[ \left( u - \frac{M}{2} + u_k \right)^2 + \left( v - \frac{N}{2} + v_k \right)^2 \right]^{\frac{1}{2}}$$

I filtri notch passanti, invece, si ottengono a partire dal filtro notch di rigetto nella stessa maniera in cui si ottiene un filtro passa alto da un filtro passa basso, ossia

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$



### Computazione della trasformata di Fourier discreta

**La trasformata di Fourier discreta è separabile**, ossia si può effettuare la trasformata di Fourier discreta monodimensionale delle righe, seguita da quella delle colonne. Così facendo risulta più semplice la sua implementazione, considerando una sola variabile per volta, e ciò **vale anche per la trasformata di Fourier discreta inversa**.

La **complessità di questa trasformata** è  $O((MN)^2)$ , con MN la dimensione dell'immagine, tuttavia **la trasformata di Fourier discreta può essere implementata anche tramite la trasformata di Fourier veloce** (Fast Fourier Transform), la quale ha una **complessità**  $O(MN \log_2(MN))$ . L'implementazione della trasformata di Fourier veloce richiederebbe che le dimensioni M e N siano potenze di 2, ma può essere generalizzata per “eliminare” questo vincolo.

Come si può notare dalla complessità, **la trasformata di Fourier veloce risulta essere molto più veloce della trasformata di Fourier discreta**.

### Filtraggio nel dominio spaziale vs nel dominio della frequenza

Considerando un'immagine di dimensione MN fissata ed un filtro di dimensione mn, **è conveniente lavorare nel dominio spaziale quando si hanno filtri relativamente piccoli** (in termini di dimensioni), con il costo della convoluzione nel dominio spaziale pari a  $O((mn)(MN))$ , mentre **è preferibile lavorare nel dominio**

**della frequenza quando si hanno filtri grandi**, con il costo del filtraggio che è uguale a quello della trasformata di Fourier veloce, ossia  $O(MN \log_2(MN))$ .

