

## Lez 1 → INTRO ELIM

Nel contesto delle elaborazioni delle immagini abbiamo diversi obiettivi:

- Manipolazione di immagini per il miglioramento dell'immagine. Poiché un dispositivo di acquisizione può avere difetti, ci sono manipolazioni per migliorarle, utili per l'interpretazione umana e per lo svolgimento di compiti più avanzati, ad esempio *computer vision*.
- Individuare automaticamente elementi caratteristici di una scena per sfruttare queste *features* per fare riconoscimento di un oggetto specifico o un volto...

L'immagini vengono manipolate anche per altri scopi: per comprimerle per scopi di trasmissione o memorizzazione ecc...

---

## ELABORAZIONE DELLE IMMAGINI

L'informatica riguarda come gestiamo ed elaboriamo le informazioni, come sono rappresentate e trasmesse.

Le info hanno diversi tipi, nel caso dell'elim l'info è sotto forma di un'immagine.

## COS'E' UN'IMMAGINE DIGITALE?

Sono immagini rappresentabili in un calcolatore elettronico. Le immagini non le acquisiamo direttamente in formato digitale ma devono essere sottoposte a un processo per renderle tali, dato che per natura sono analogiche.

Si applica un procedimento per digitalizzare l'immagine.

L'immagine è un concetto specifico di un segnale. Quando acquisiamo immagini subiscono un processo di trasformazione che in un certo momento sono rappresentate da un segnale analogico dove le grandezze hanno valore numerico di tipo reale.

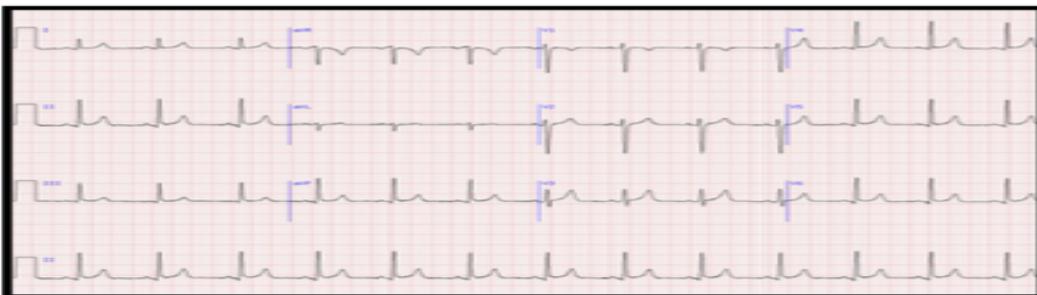
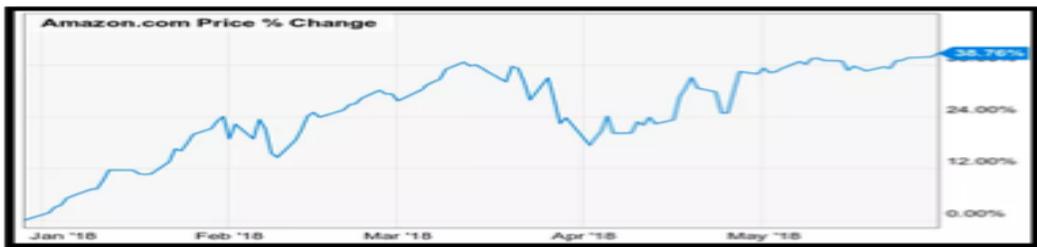
Ciò significa che un'immagine così com'è non è rappresentabile con un calcolatore. Le immagini devono quindi essere convertite con un processo chiamato *discretizzazione*.

---

## SEGNALI

Quando abbiamo a che fare col mondo fisico trattiamo con dei segnali.

Un segnale è una funzione che dipende da una variabile che può essere il tempo o lo spazio.



Il primo grafico rappresenta le fluttuazioni di Amazon ed è un segnale 1D.  
Il secondo esempio è un elettrocardiogramma.

**Una qualsiasi quantità che varia nel tempo o nello spazio è un segnale.**

Abbiamo poi funzioni 2D, ad esempio longitudine e latitudine di un palazzo in città...

---

## SEGNALI ANALOGICI E DIGITALI

Per studiare i segnali devo imparare ad acquisire il segnale.

La figura mostra l'acquisizione di un segnale che in questo caso è la voce.

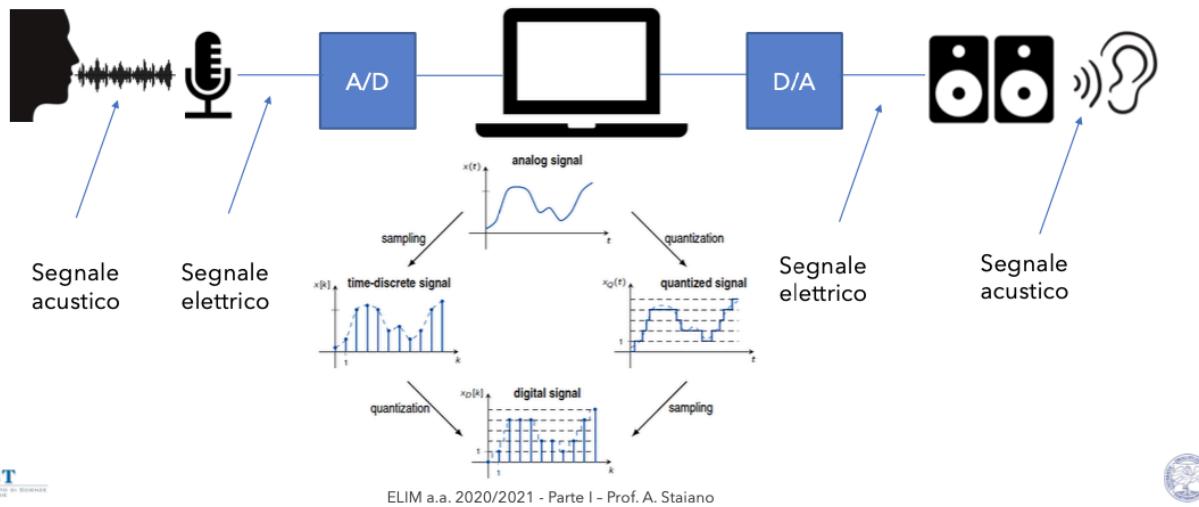
Quando devo elaborare un segnale acustico ho la voce che provoca delle vibrazioni delle molecole d'aria che vengono catturate da un microfono, il microfono converte le vibrazioni in un segnale elettrico, dopo che abbiamo il segnale elettrico dobbiamo discretizzarlo (convertire da analogico a digitale) e questo è il compito di un convertitore A/D (analogico/digitale). L'A/D fa un lavoro importante: prima dell'AD abbiamo un segnale elettrico (analogico) che varia nel tempo e assume dei valori continui. Per rappresentarlo in un calcolatore dobbiamo discretizzarlo. Ciò avviene con due operazioni dell'AD:

- Il segnale viene discretizzato sia rispetto alla var indipendente (CAMPIONAMENTO) sia a quella dipendente (QUANTIZZAZIONE). Queste due operazioni discretizzano il segnale.

Il campionamento esamina il segnale e osserva il valore in punti equi spaziati e fissati della var indipendente e li considero solo in quei punti.

L'intensità del segnale è comunque un valore continuo e deve essere discretizzato con la quantizzazione: attribuire a un intervallo dei possibili valori della var dipendente un valore specifico.

Il risultato finale è il segnale discretizzato, fatto ciò, posso usarlo in un calcolatore.



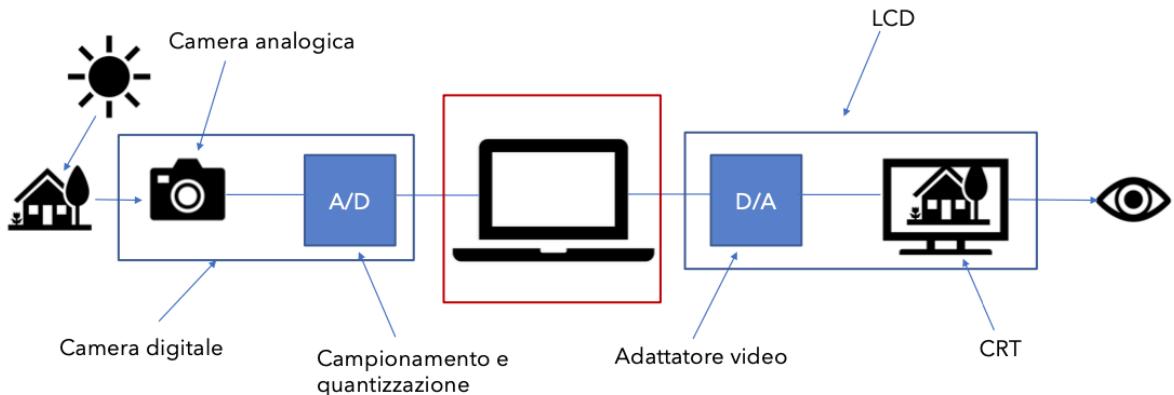
Dopo aver discretizzato devo vedere l'effetto creato, quindi c'è bisogno di un convertitore D/A che ritrova il segnale elettrico e grazie a un altro dispositivo (casse) possiamo ascoltare il risultato finale.

Lo stesso avviene per le immagini: il dispositivo di acquisizione è un dispositivo di *imaging* che è in grado di convertire le onde luminose in un segnale elettrico.

Abbiamo una scena acquisita da una macchina fotografica che è un dispositivo che cattura i fotoni che provocano dei processi fotochimici che causano reazioni chimiche che vanno impresse sulla pellicola, dopodiché l'immagine deve andare in input a un A/D che discretizza l'immagine.

I moderni dispositivi di imaging fanno già questo processo in un solo passo (acquisizione e conversione): le camere moderne hanno dispositivi CCD che catturano fotoni provenienti da una scena che vengono elaborati e producono un segnale elettrico che viene discretizzato. Alla fine abbiamo un segnale digitale rappresentato in un calcolatore che può essere manipolata.

## SEGNALI ANALOGICI E SEGNALI DIGITALI



Dopo aver manipolato l'immagine posso fare il processo inverso per visualizzarla.

I dispositivi moderni hanno una scheda video che effettuano automaticamente la conversione finale

---

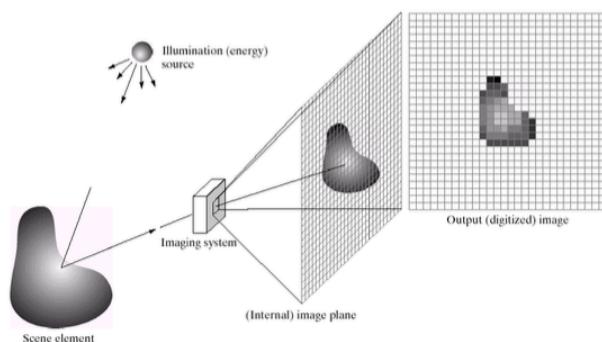
## IMMAGINI DIGITALI

Dopo la discretizzazione come rappresentiamo l'immagine?

L'immagine è una matrice in cui abbiamo le coordinate  $x, y$  dei punti dell'immagine e ad ogni punto corrisponde un valore di intensità. Di fatto è una funzione 2D discreta.

Ciascun punto prende il nome di pixel. Più pixel ho e più l'immagine è accurata e definita. Non è solo quello che definisce la qualità dell'immagine ma anche i valori del livello di grigio che posso rappresentare, più sono meglio è.

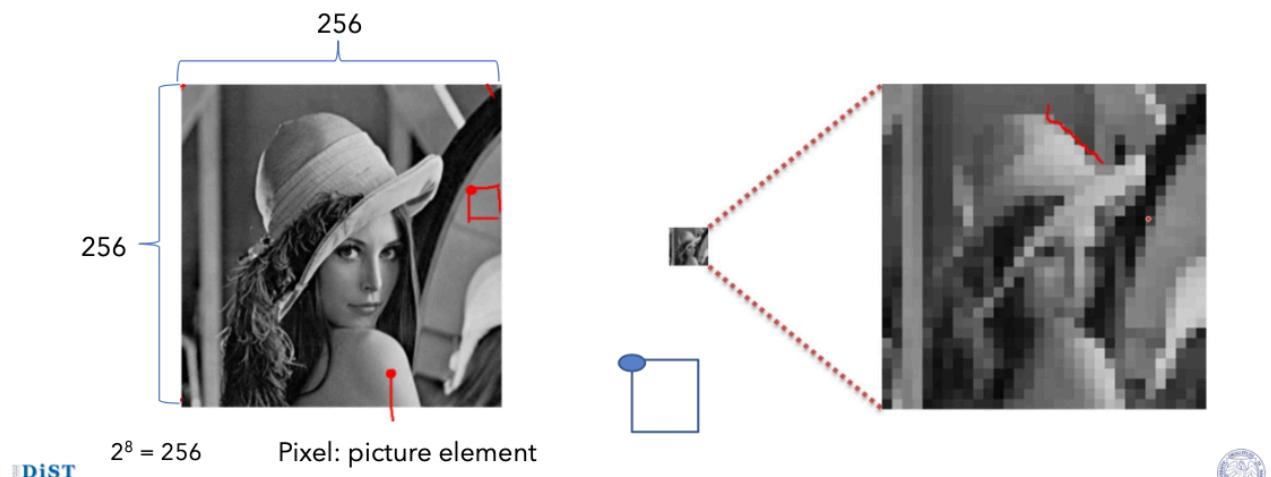
Quando carichiamo immagini in un calcolatore è una matrice  $m \times n$  dove in ciascun punto c'è un valore di intensità. Normalmente i valori sono 256 e sono sufficienti 8bit ma possiamo usare sia un numero superiore che inferiore e quindi i bit varieranno.



## CAMPIONAMENTO (SAMPLING)

Quando diciamo che l'immagine deve essere discretizzata avvengono le operazioni di quantizzazione e campionamento.

Per campionamento prendiamo un certo numero di punti dall'asse delle  $x$  (*var indip*) e più punti ho e più pixel avrò.



Questa immagine è 256 x 256: supponiamo di fare un campionamento di questa immagine e prendiamo 1 pixel per ogni 8, ottenendo una dimensione di 32 x 32. Come vediamo l'effetto del sottocampionamento, usando meno pixel la qualità dell'immagine precipita.

## QUANTIZZAZIONE

Lo stesso avviene per la quantizzazione: abbiamo la rappresentazione con 256 livelli di grigio, poi con 16 e già vediamo differenze, soprattutto se ne usiamo 4...

Se abbiamo immagini con molti pixel e molti livello di grigio avremo un'immagine molto pesante in termini di memoria e la quantizzazione ci aiuta a capire le giuste misure da adottare.

## QUANTIZZAZIONE



## TIPOLOGIA DI ELABORAZIONI NELLA IMMAGINI DIGITALI

Quando si parla di elim si può fare confusione tra diversi termini:

- elim
- Computer vision
- Comprensione delle immagini

Ma sono tutte attività che permettono la realizzazione comune di attività.

Di solito sono riconosciute 3 categorie di elaborazione, caratterizzate dal loro livello di astrazione:

- Basso livello (miglioramento di immagini)
  - Medio livello (estrazione di caratteristiche o feature)
  - Alto livello (riconoscimento di oggetti)
- Miglioramento delle immagini: modificare la foto per renderla più pulita, manipolando i pixel, parliamo di basso livello.
  - Per i compiti a medio livello abbiamo l'obiettivo di individuare caratteristiche peculiari del contenuto delle immagini
  - Per i compiti ad alto livello ci avviciniamo alla CV come la classificazione degli oggetti.
- 

## DIFFERENZA TRA ELIM, CV E ELABORAZIONI DELLE IMMAGINI?

Dipende dal livello di astrazione che dobbiamo svolgere.

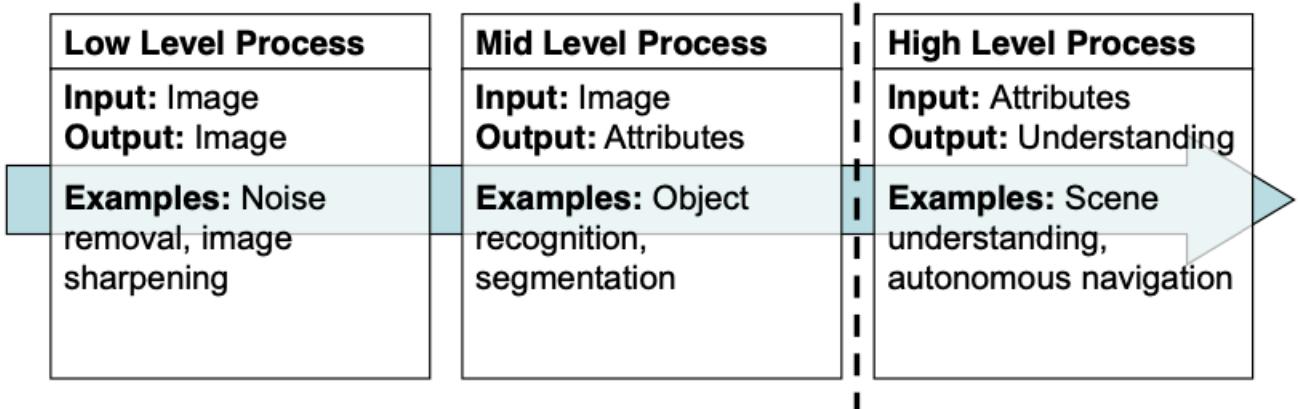
La CV si occupa del mondo 3D: parte da un video o immagine per ricostruire una scena 3D: entrano in gioco caratteristiche geometriche e dinamiche degli oggetti. Si tratta di fare inferenza sul mondo 3D.

L'elim ha a che fare con operazioni di basso livello.

Possiamo vedere le 3 operazioni come un continuo: da operazioni a basso arriviamo ad alto livello.

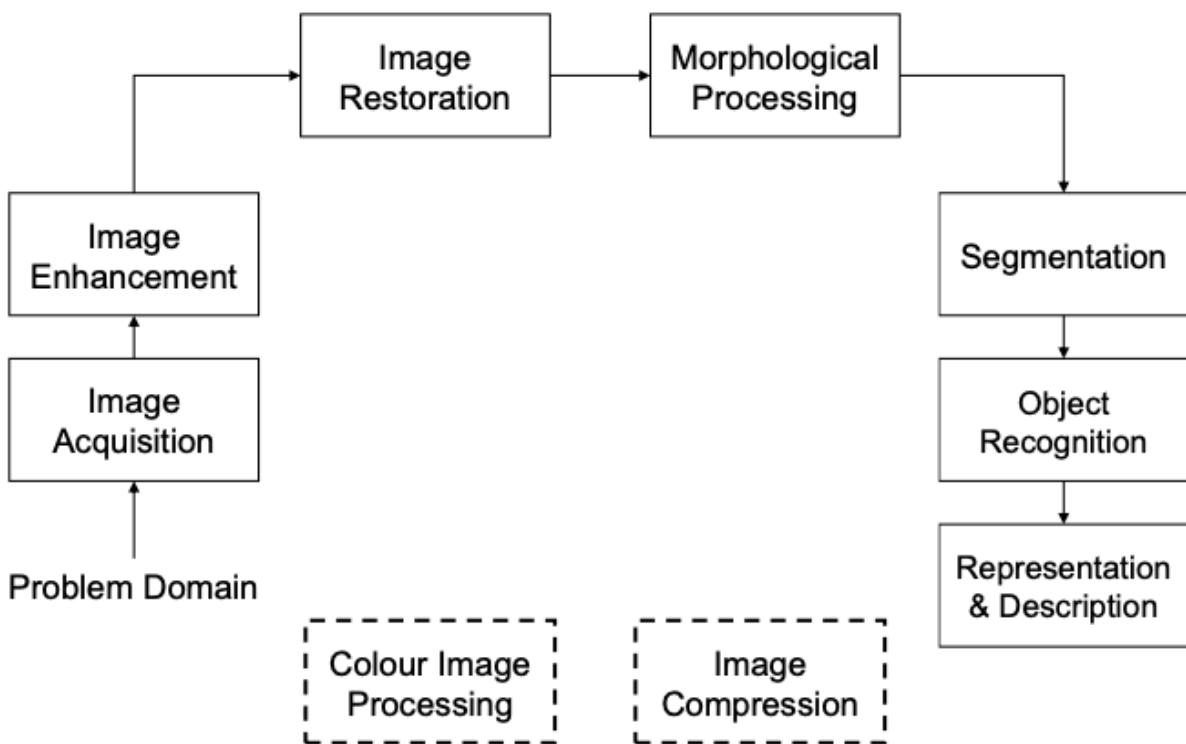
Nel caso di operazioni a basso livello l'input e l'output è un'immagine.

Nel caso di operazioni ad alto livello l'input è un video o immagine e l'output è la comprensione di ciò che sta avvenendo.



ELIM a.a. 2020/2021 - Parte I - Prof. A. Staiano

Nell'elim abbiamo diverse attività:

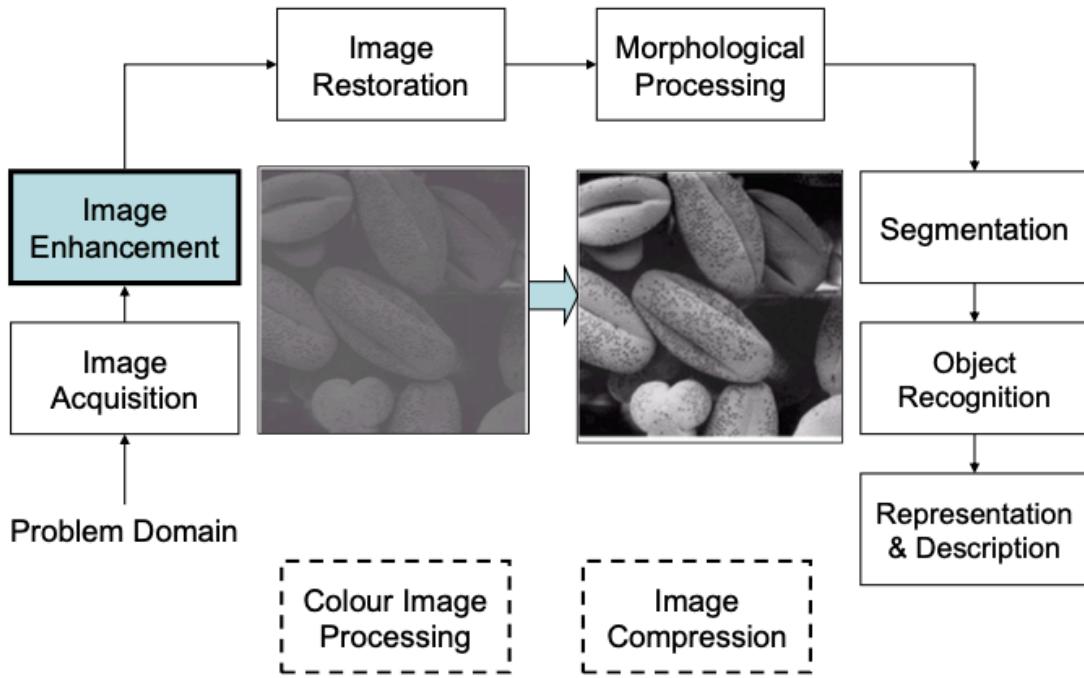


ELIM a.a. 2020/2021 - Parte I - Prof. A. Staiano

## FASI CHIAVE: MIGLIORAMENTO IMMAGINI

Abbiamo operazioni pixel per pixel e regioni di pixel.

Cosa si intende migliorare: prendere un'immagine, manipoliamo per migliorare l'aspetto o per un'interpretazione umana o per un calcolatore.



ELIM a.a. 2020/2021 - Parte I - Prof. A. Staiano

**RECUPERO IMMAGINI:** (non faremo molto) → si basa su modelli di degrado matematici che migliorano le immagini in maniera oggettiva.

**ELABORAZIONE MORFOLOGICA:** ha a che fare con l'estrazione di caratteristiche dalle immagini che si applicano su immagini principalmente in bianco e nero.

**SEGMENTAZIONE:** ho un'immagine che contiene diversi oggetti nella scena e determiniamo i contorni di ciascun oggetto per poterli distinguere.

Dopo aver fatto ciò possiamo fare riconoscimento di oggetti.

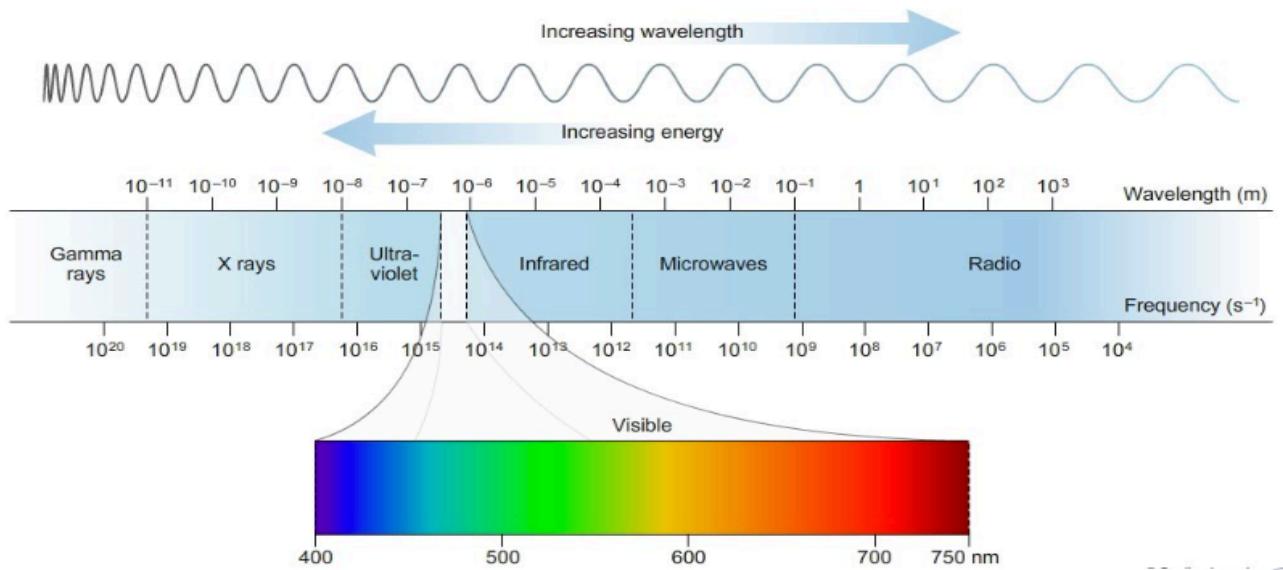
---

### TIPI DI IMMAGINI E SPETTRO ELETTRONAGNETICO

Fino ad ora abbiamo detto che possiamo acquisire immagini digitalizzate e portate in un calcolatore.

Lo spettro elettromagnetico è la rappresentazione di tutte le onde elettromagnetiche sulla base della loro frequenza. Ne esistono diverse tipologie di frequenza. Quelle che siamo in grado di percepire con i nostri occhi è una

piccolissima quantità. Le onde partono da quelle radio ai raggi gamma. Le onde con maggior frequenza trasportano più energia.



Le onde a bassa frequenza sono quelle più lunghe. Le varie onde hanno altre proprietà a seconda della loro frequenza, ad esempio le onde radio sono onde lunghe che attraversano anche gli ostacoli, al contrario delle onde ad alta frequenza.

Siamo in grado solo di percepire una minima quantità (quella in bianco in foto). Le nostre macchine fotografiche lavorano con la suddetta banda ma esistono dispositivi che generano immagini anche in altre bande e hanno applicazioni specifiche come medicina, astronomia...

Le immagini non provengono solo da onde elettromagnetiche ma anche da quelle acustiche o elettroniche o sintetiche.

## CLASSIFICAZIONE DELLE IMMAGINI

Sono classificate dalla loro sorgente.

- immagini di riflessione: catturano l'onda riflessa dall'oggetto.
- Immagini di emissione: sono quelle ottenute dagli infrarossi o da risonanze magnetiche.
- Immagine di assorbimento: raggi X e gamma.

## **IMMAGINI A RAGGI-GAMMA**

Le immagini che possiamo elaborare non sono per forza derivanti da una fotografia di una macchina fotografica ma da un qualsiasi dispositivo che opera nelle bande dello spettro elettromagnetico.

Le applicazioni dei raggi gamma sono concentrate nell'astronomia e nella medicina.

---

## **IMMAGINI A RAGGI-X**

Abbiamo diverse applicazioni: campo industriale, medico.

---

## **IMMAGINI AD ULTRAVIOLETTO**

Spesso è sfruttato il fenomeno della fluorescenza.

---

## **IMMAGINI A LUCE VISIBILE**

Sono quelle percepite dall'essere umano. Ci sono molte applicazioni.

## **IMMAGINI A LUCE VISIBILE E A INFRAROSSO**

Sono utili per studi riguardanti il *remote sensing*: immagini prese da satelliti e sono adatte a rappresentare zone geografiche.

Le immagini multispettrali usano diverse bande di freq.

---

## **IMMAGINI A INFRAROSSO DA SATELLITE**

Sono usate per studiare anche l'attività antropica delle città.

## **IMMAGINI DA SATELLITE RADAR**

Le microonde possono passare attraverso le nuvole per avere un'immagine pulita. Es catena del Tibet sempre nuvolosa e grazie a questa applicazione possiamo ricostruire la superficie per avere l'immagine fedele della catena montuosa.

---

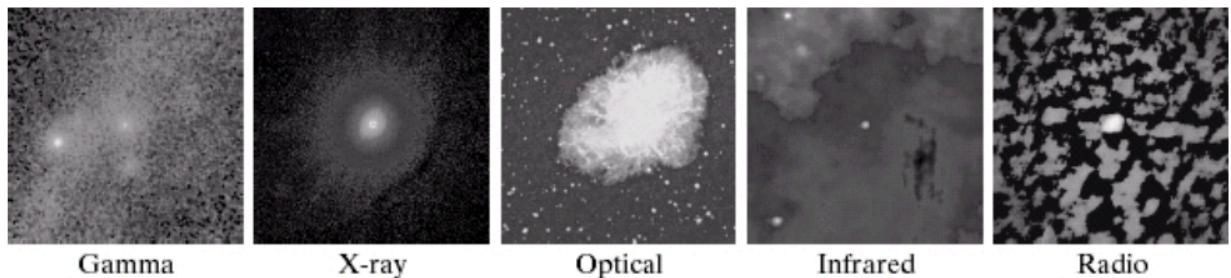
## **IMAGING AD ONDE RADIO**

Sono usate per le risonanze magnetiche.

---

## **IMAGING MULTISPETTRALE**

Rappresentazione di uno stesso oggetto ripreso in bande diverse.



Ogni onda dà informazioni differenti.

---

### IMMAGINI AD ONDE ACUSTICHE

Ci sono immagini anche prodotte da onde acustiche: questo tipo di applicazione è usato per lo studio dei fondali marini: sono inviate onde che quando toccano il terreno ritornano e a seconda del tempo di ritorno consente di ricostruire il fondale.

---

### IMMAGINI AD ULTRASUONI

Molto usate per le ecografie.

### IMMAGINI A FASCIO DI ELETTRONI

Ci sono immagini prodotte da microscopi elettronici che usano un fascio di elettroni che toccano lo schermo e crea l'immagine ingrandita.

---

### QUALI SONO LE APPLICAZIONI DELL'ELIM DIGITALI?

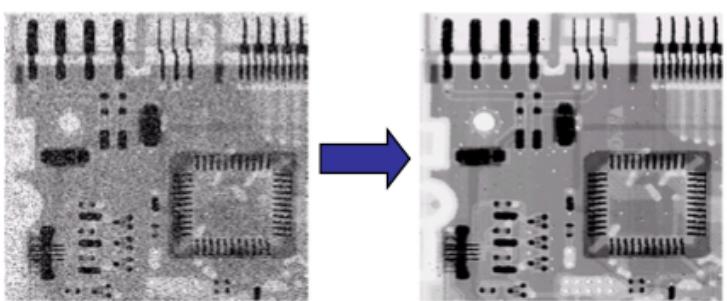
Possiamo avere a che fare con molte immagini provenienti da diverse fonti energetiche.

I campi applicativi sono molteplici: medicina, astronomia, sicurezza, industriale, topografia...

---

### Esempio: MIGLIORAMENTO DELLE IMMAGINI

Usiamo tecniche o per migliorare il contrasto o per eliminare il rumore.

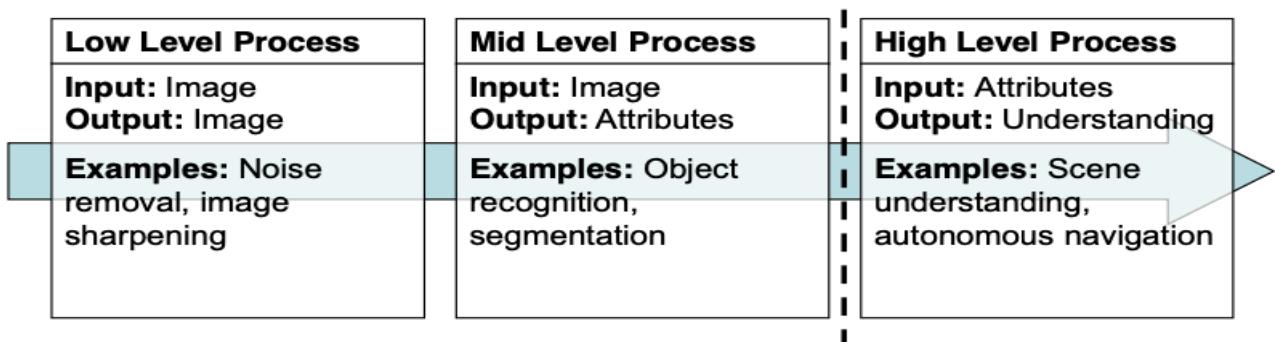


Eliminando il rumore ottengo un'immagine più definita.

## Lez 2 → Introduzione all'AI e ML

C'è una sovrapposizione tra ELIM e AI, sia con la segmentazione delle immagini (esaminare l'immagine e contornare i soggetti della scena) che compiti ad alto livello.

La sovrapposizione si ha a partire dai compiti a medio livello di astrazione dove diamo un'immagine in input ed estraiamo proprietà associate ad essa.



Le tecniche classiche di segmentazione che si adottano prevedono di fare contorno di un soggetto senza tener conto dell'oggetto, la tendenza degli ultimi anni è cercare di fare questa operazione, riuscendo ad identificare anche l'oggetto (animale, uomo...) e per fare queste operazioni ci si basa su conoscenze di IA, ad esempio reti neurali, ciò è detta segmentazione *semantica*. Ancora più chiaro è l'uso per i compiti ad alto livello: si usano solamente tecnologie IA.

Ecco il motivo per il quale bisogna avere un'idea di AI e ML.

### COS'E' L'INTELLIGENZA ARTIFICIALE?

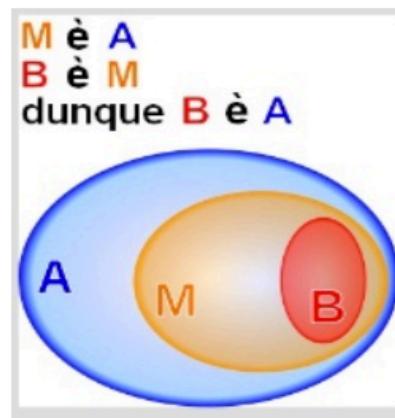
*È un'intelligenza contrapposta a quella naturale, prerogativa degli esseri viventi.*

È una disciplina che ha origine dopo la 2 guerra mondiale ed è cresciuta grazie al contributo della fisica, informatica, psicologia, filosofia...

Già Aristotele formulò un insieme preciso di leggi che governano la parte razionale della mente, stilando dei sillogismi per un corretto ragionamento che permettono di generare conclusioni meccanicamente, dato un insieme di dati di partenza.

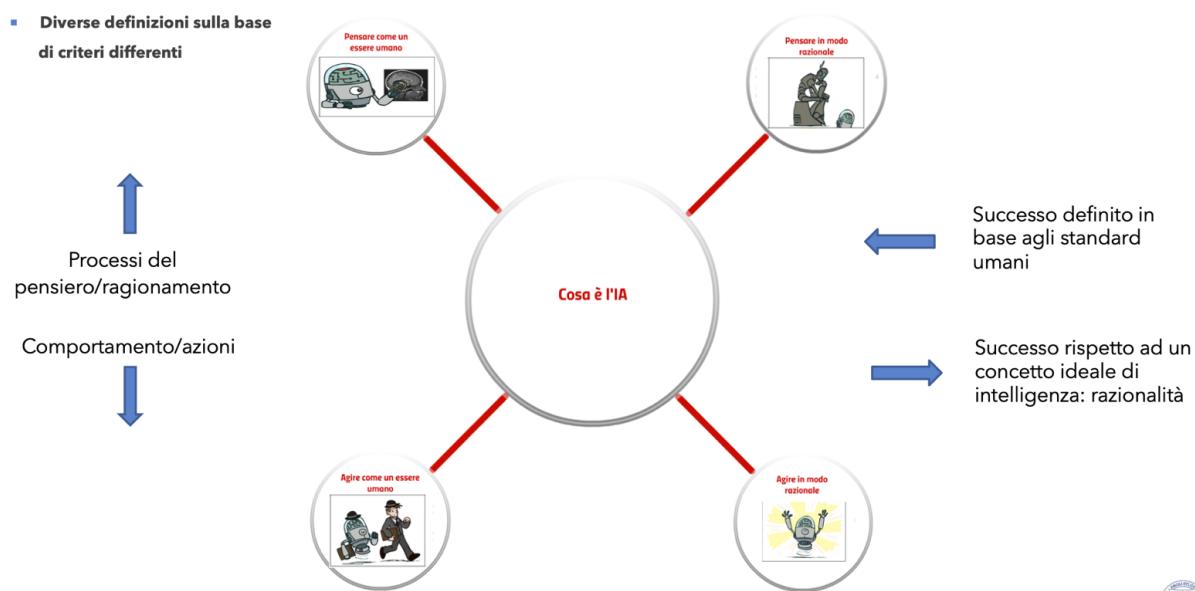
- Esempio

- Tutti gli uomini (M) sono mortali (A)
- Socrate (B) è un uomo (M)
- Socrate è mortale (A)



## DEFINIRE L'INTELLIGENZA

1. Per stabilire se qualcosa è intelligente per prima cosa si tiene come riferimento il comportamento umano.
2. Pensare in modo razionale: in base alle conoscenze prendiamo sempre una decisione “giusta”.
3. Focus sul comportamento: cosa faremmo noi in una certa situazione?  
Sviluppare sistemi artificiali e confrontarli col comportamento umano
4. Focus sull’azione in modo razionale.



## PENSARE COME UN UMANO

Le definizioni che sono incentrate sul pensiero umano, sono nate da una cosa: se devo creare un sistema che pensa come un umano, dobbiamo prima capire come pensa un umano... in questa impostazione prima si costruisce una teoria della mente e poi la si riproduce in un sistema.

Questo approccio parte della psicologia e dallo studio delle scienze cognitive.

---

## AGIRE COME UN UMANO

Dobbiamo fare il confronto con quello che farebbe un essere umano. Su questa impostazione si basa quello che è noto come ***Test di Turing***.

---

### ALAN TURING

Alan Turing era un matematico che ha dato molti contributi all'informatica. Era affascinato dalle macchine, si è posto poi il problema di avere una macchina pensante.

Concepì il Test di Turing: si basa sul gioco dell'imitazione. Turing sostituì la donna col computer: un interrogatore che pone domande, un umano che risponde e un computer che deve rispondere in modo da ingannare l'interrogatore, se riesce a far ciò, il computer è intelligente.

Una macchina per essere intelligente deve avere molte competenze: deve comprendere il linguaggio umano, avere una fonte di ragionamento, deve essere in grado di rappresentare la conoscenza applicando dei ragionamenti e saper apprendere nozioni fornite.

---

## PENSARE IN MODO RAZIONALE

Cercare di riprodurre i meccanismi del pensiero. Questi studi hanno dato origine al settore della Logica: tramite essa è possibile riprodurre il ragionamento umano.

---

## AGIRE IN MODO RAZIONALE

In breve, fare la cosa giusta. Quantifichiamo obiettivamente il costo delle azioni e quando abbiamo un obiettivo massimizziamo il costo.

Il vantaggio è avere un metodo oggettivo per fare un confronto per capire quanto il sistema sviluppato è intelligente.

---

## FONDAMENTI DELL'IA

Filosofia, matematica, economia, linguistica, neuroscienze, teoria del controllo sono tutte discipline fondanti di questa materia.

---

## CRONISTORIA DELL'IA

Tutto nasce tra gli anni '40-'50. Uno dei primi lavori è datato 1943 dove 2 studiosi si basarono su neuroni e logica hanno sviluppato un modello di neurone artificiale: avevano dimostrato che con esso si potevano svolgere semplici funzioni booleane.

Nel 1950 con Turing si ebbe la prima visione piena dell'IA.

**In Italia, Eduardo Caianiello (Napoli), fu uno dei primi studiosi di reti neurali negli anni '60.**

La disciplina dell'IA è nata formalmente nel 1956.

Tutto nacque con un Workshop che riunì le migliori menti sulla teoria degli automi e reti neurali.

Essi ritenevano di poter formalizzare e costruire macchine intelligenti e gettarono le basi per capire come agire.

Venne definito il termine AI per la prima volta.

Per i successivi 20 ci furono moltissimi progressi, tanto da usare il termine “Look, Ma, no hands” a sottintendere la facilità con cui si progrediva in questa nuova disciplina.

Ovviamente erano problemi molto banali.

Tra gli anni '66-'73 vi fu un collasso nella ricerca:

- Progressi molto più lenti rispetto alle attese.
- Alcuni sistemi non scalabili.
- Limiti nelle rappresentazioni e tecniche.

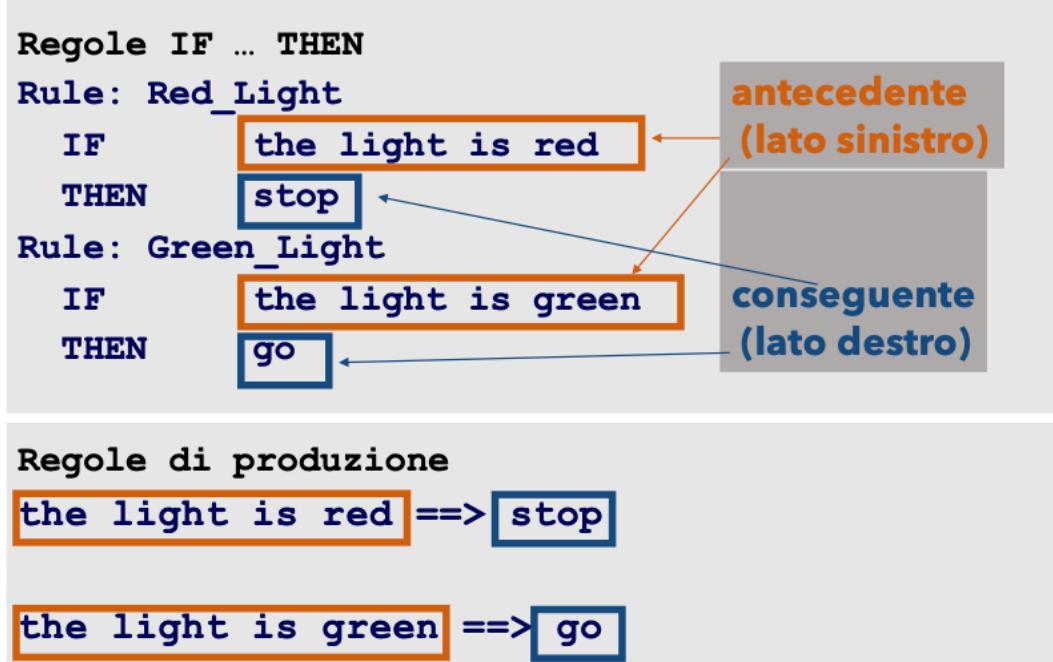
Le cose si orientarono poi verso **sistemi esperti**: sono una categoria di sistemi dove codifichiamo della conoscenza sul dominio applicativo mediante un insieme di regole, il sistema è dotato di un motore che sfruttando delle reali condizioni, attiva la parte condizione delle regole per intraprendere un'azione (IF-THEN). Tramite questo processo inferenziale i sistemi erano in grado di prendere decisioni sullo specifico dominio applicativo.

Un sistema esperto è costituito da una base di conoscenza (fatti noti di un dominio applicativo).

C'è poi un motore che in base alle conoscenze e a input prende decisioni che vengono valutate da esperti del settore.

---

## ESEMPIO DI REGOLE



## MYCIN

Uno dei sistemi esperti più famosi è MYCIN: un sistema usato per diagnosi batteriologiche, esaminava il sangue e dava delle conclusioni cerca la possibile malattia del paziente.

Questo sistema fu provato con medici, professori, studenti e codificarono un'ottantina di casi, dando risposte più accurate rispetto a tutti gli umani usati nell'esperimento.

MYCIN era in grado di rappresentare anche situazioni di incertezza, associando un grado di probabilità.

Questi sistemi hanno però due problemi:

- C'è bisogno di un umano che contribuisca alla base della conoscenza
- I sistemi non avevano capacità di apprendimento, determinavano conclusioni solo grazie alle cose presenti nella base di conoscenza.

Ecco perché questi sistemi non sono molto usati ormai, se non in campi specifici.

---

## BREVE STORIA DELL'IA

Negli anni '80 l'IA iniziò a diventare un'attività commerciale.

La crisi dell'IA finì nel 1986 dove si sviluppò un algoritmo che dotava una rete neurale la capacità di apprendimento.

Nel 1987 l'IA divenne una scienza e si formarono due divisioni:

- **Neats**: i “precisini” → uso della matematica ecc...
- **Scruffies**: gli “scapigliati” → provare e vedere come va.

La verità è nel mezzo: nel 2011 prende l'avvento il *deep learning* dove c'è una base solida ma si lavora come gli *Scruffies*, senza sapere perché funziona.

---

## PROGETTARE AGENTI RAZIONALI

Un agente è un'entità che percepisce ed agisce, possiamo immaginarlo come un robottino dotato della capacità di percepire l'ambiente (sensori tattili e visivi), saper interpretare e intraprendere azioni che massimizzano una funzione obiettivo.

---

## COSA PUO' FARE OGGI L'IA?

- ✓ □ Giocare una buona partita a ping-pong?
- ✓ □ Giocare una buona partita a Jeopardy?
- ✓ □ Guidare in modo sicura su una strada di montagna piena di curve?
- ✗ □ Guidare in modo sicuro lungo Corso Meridionale?
- ✓ □ Fare la spesa di una settimana sul web?
- ✗ □ Fare la spesa di una settimana al mercato?
- ✗ □ Scoprire e dimostrare un nuovo teorema di matematica?
- ✗ □ Conversare correttamente con una persona per un'ora?
- ✗ □ Eseguire un'operazione chirurgica?
- ✓ □ Mettere via i piatti e piegare la biancheria?
- ✓ □ Tradurre da Cinese a Inglese in tempo reale?
- ✗ □ Scrivere un storia volutamente divertente?

L'AI è usata per molti tipi di automazione:

- Ricerca web
  - Classificatore di mail spam
  - Consigli su prodotti
  - Pianificazione di percorsi
-

## MACHINE LEARNING

È un sottoinsieme dell'area dell'IA.

L'idea è dotare i computer della capacità dell'uomo e degli animali: apprendere dall'esperienza.

Come funziona una rete neurale? Avendo a disposizione un insieme di immagini di gatti e cani, dato un input la rete deve distinguere gli animali.

Come fa ad apprendere? Con l'esperienza: per prima cosa la addestriamo mostrandogli  $n$  gatti e cani e sulla base di ciò imparerà a distinguerli.

C'è quindi la *fase di addestramento* e più dati abbiamo meglio si apprende.

Questa fase dura finché non si ritiene una buona qualità di apprendimento.

Inizia poi la fase di *test*: prendo immagini esclusive e la macchina deve rispondere riconoscendo i soggetti, se l'apprendimento è stato efficiente allora raggiungeremo addirittura il 100% di accuratezza.

Queste tecniche sono dette **supervisionate** poiché nella prima fase c'è un essere umano che addestra la macchina ma ci sono applicazioni senza supervisione.

Nell'apprendimento supervisionato ci sono due tecniche: **classificazione** e **regressione**:

- La **classificazione** predice risposte discrete, ad esempio se una mail è spam o meno. Nella classificazione l'obiettivo è predire un output che ha valori discreti.
- La **regressione** predice risposte continue, ad esempio cambiamenti della temperatura o fluttuazioni delle richieste di energia.

Nell'apprendimento non supervisionato non c'è nessuna risposta giusta nei dati che vengono forniti.

Una tecnica usata è il **clustering**: diamo in input dei dati e l'algoritmo deve individuare delle similitudini all'interno di questi dati (individua dei gruppi con similitudini comuni che li differenziano dagli altri partecipanti).

---

## Lez 3 → ML: APPRENDIMENTO SUPERVISIONATO

Dopo aver introdotto alcuni concetti di IA, abbiamo parlato del ML, dicendo che ML fa riferimento a un insieme di tecniche in cui la parola chiave è **apprendimento**. Ci sono metodologie in base alle quali riusciamo a determinare un comportamento che non deve essere programmato in modo esplicito ma l'apprendimento è determinato dall'esperienza che è costituita da dati. Ecco perché diciamo che le tecniche di ML apprendono dai dati. Per poter realizzare attività con ML abbiamo bisogno di **dati**.

Questo apprendimento avviene in diversi paradigmi:

- **Supervisionato** (più diffuso).
  - **Non Supervisionato**.
- 

### APPRENDIMENTO SUPERVISIONATO: ESEMPI

In qualsiasi tipo di attività prima di elaborare i dati, devo conoscerli facendo una serie di grafici che ci danno info sulle loro caratteristiche poiché i dati possono essere sempre visualizzati nello spazio cartesiano.

In questo caso siamo in un'applicazione medica, in particolare uno studio dove 97 pazienti con problemi prostatici. Date una serie di informazioni per ciascun paziente (volume del cancro, età...), predire il livello di PSA (parametro prostatico).

---

### ALTRO ESEMPIO: RILEVARE EMAIL SPAM

Dati 4061 mail, ognuna è etichettata come spam o e-mail.

L'obiettivo è costruire un filtro di spam.

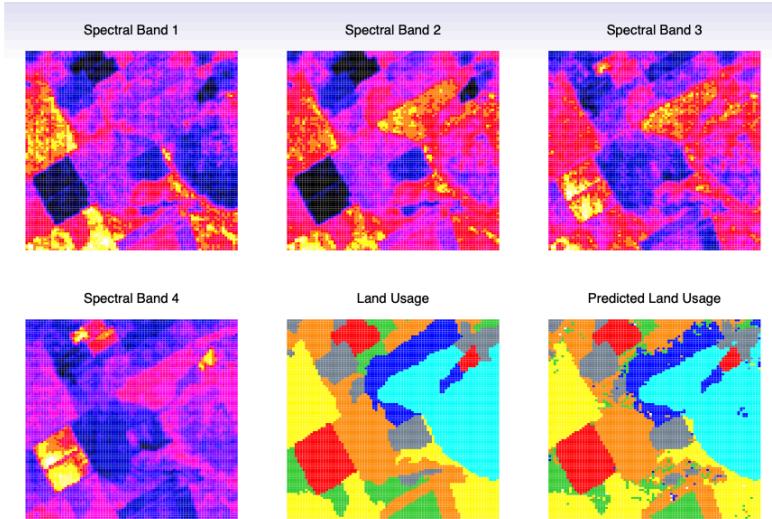
In input: 57 frequenze che occorrono in tali mail.

	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

Anche qui abbiamo un problema di predizione: date una serie di info circa delle mail di cui sappiamo frequenze e etichetta (mail o spam), creare un filtro antispam classificando le mail in entrata.

## ESEMPIO 3: IDENTIFICARE I PIXEL DI UN'IMMAGINE

Abbiamo immagini prese da un satellite in diverse bande di frequenza e sulla base di queste info, l'obiettivo è classificare le immagini.



## APPRENDIMENTO SUPERVISIONATO

Gli esempi citati sono diversi ma hanno cose in comune:

- Hanno dati con parametri di input e più output.
- **L'obiettivo** è usare gli input per predire gli output.

Per mettere su un sistema per far apprendere i modelli è necessario che per ciascun dato in input che passiamo, dobbiamo indicare qual è l'output corrispondente:

- Nel caso dei pazienti PSA dobbiamo indicare qual è il valore di PSA di ciascun paziente.
- Nel secondo caso: oltre alla mail dobbiamo anche dire se è spam o mail
- Nel terzo caso dobbiamo specificare a quale tipologia appartiene.

Tutto ciò è chiamato **apprendimento supervisionato**: l'info sull'output corretto è supervisionato dall'utente e sulla base di esso la macchina deve imparare a riconoscerli su dati futuri non noti.

## VARIABILI DI OUTPUT QUANTITATIVE E QUALITATIVE

Gli output possono avere diverse tipologie:

- Nel caso del PSA devo predire un valore continuo, è quantitativo.

- Negli altri due casi l'output è qualitativo (**discreta**): appartiene a un insieme di categorie dove non c'è un ordinamento predefinito. Non è una variabile continua perché è una categoria che rappresentiamo con codifiche numeriche.

La differenza nella tipologia della var di output determina il tipo di problema da risolvere:

- se è continua parliamo di problema di **REGRESSIONE**
- se è discreta il problema prende il nome di **CLASSIFICAZIONE**.

Nel gergo del ML la variabile di output è anche chiamata **target** (obiettivo da imparare a predire).

Nel caso della classificazione, dato che dobbiamo individuare una categoria di appartenenza, la variabile è chiamata **label**.

Regressione e classificazione sono problemi di predizione: dato un input il sistema deve imparare a predire, dal punto di vista matematica è un problema di **approssimazione**.

Quando ho dei dati da imparare, nel gergo quell'insieme è chiamato *dataset*. Questi dati devono essere anche organizzati: una parte usati per l'addestramento della macchina (*training-set*) e per vedere se ha imparato il modello devo usare un altro insieme di dati esclusivo (*test-set*).

Ovviamente in tutti i dataset i dati di input (ad esempio i pazienti) sono caratterizzati da un insieme di variabili, e anche per le var di input abbiamo diverse tipologie: è possibile avere situazioni per le quali anche le var di input hanno valori discreti e vanno codificate opportunamente (supponiamo che il nostro classificatore deve classificare le mail spam o meno [0 o 1] ma se abbiamo più categorie posso associare  $n$  valori, ma in realtà possiamo usare una codifica binaria e si mette il valore 1 alla classe corrispondente mentre tutti gli altri sono 0 (esempio immagini dei satelliti) questa codifica è chiamata **one-hot**.

## NOTAZIONI

Quando abbiamo un insieme di dati, viene chiamato  $X$ .

$X$  rappresenta un vettore di input: significa che  $X$  è associato a ogni paziente e per ognuno di esso deve contenere tutti gli input associati (età, dimensione tumore...). In fase di addestramento dobbiamo avere info sull'output associato:

se è un problema di regressione lo chiamiamo  $Y$  se è classificazione è chiamato  $C$ .

Il nostro training-set è un insieme di coppie  $(X_i, Y_i)$  dove  $X_i$  è il vettore di input associato al singolo dato di input (campione) e  $Y_i$  (ma anche  $C_i$ ) rappresenta il valore target (valore che il sistema deve imparare a predire, l'output corretto).

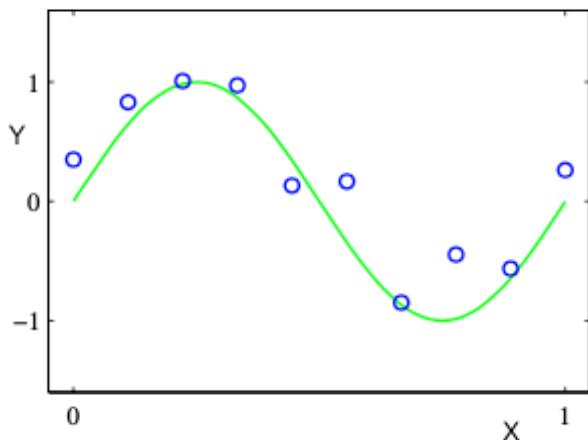
## REGRESSIONE

Supponiamo di avere i valori di una variabile di input  $X$  e di voler usare questi valori per predire i valori della variabile di output  $Y$ .

L'obiettivo è trovare una funzione che approssimi accuratamente i dati del **training-set**.

Nell'esempio il sistema dovrà imparare una funzione sinusoidale.

Nell'esempio la variabile di input è continua (regressione).



La scelta del modello da usare è chiamato **model selection**.

Ogni modello di ML ha dei parametri.

Ad esempio nel caso del polinomio devo imparare i coefficienti. Rappresento l'output della funzione in funzione di  $X$  e di  $w$  (parametri). Siccome conosco l'output nel training-set mi calcolo l'output del modello e lo confronto con quello corretto, l'obiettivo è che la *funzione di errore* sia prossima allo zero, in modo che il modello predice un valore approssimativamente corretto.

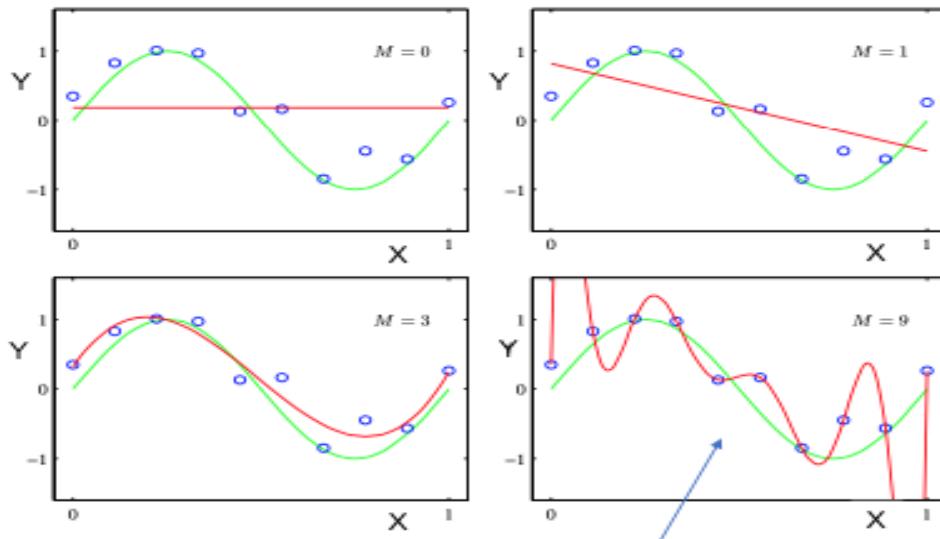
Ogni volta che viene fatto l'addestramento mi calcolo la soluzione d'errore e vado a modificare i pesi.

Nella classificazione il concetto è uguale, anche se è più adeguata un altro tipo di funzione di errore.

## MODEL SELECTION

Supponiamo di avere il problema precedente e scegliamo un polinomio:

- supponiamo innanzitutto di usare un polinomio di grado 0, in quel caso determinerei una retta che non è corretta.
- Lo stesso vale per un polinomio di grado 1.
- Posso provare che col polinomio di grado 3 fa un buon lavoro: non è esattamente preciso ma grosso modo approssima bene.
- Usando un polinomio di grado 9, sui dati di training apprende perfettamente ma se vediamo la funzione rossa vediamo che non riflette assolutamente il flusso di dati. Questo problema è chiamato *overfitting*.



Ovviamente la qualità della scelta è influenzata anche dal numero di dati che diamo in input.

## TEST-SET

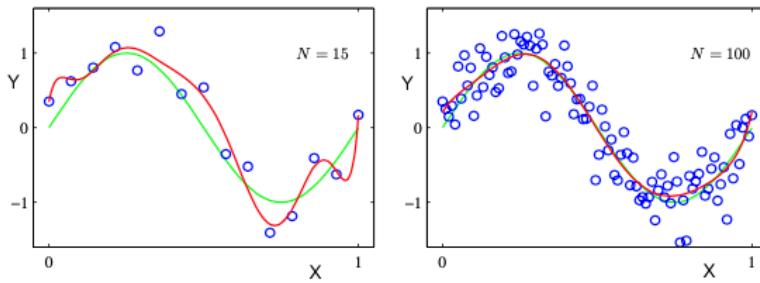
Per capire se il modello generalizza bene, si usa un altro insieme di dati, il test-set che è l'insieme col quale capisco come generalizza il modello. Abbiamo dati in input senza dare gli output al sistema per vedere come ha imparato.

Analizzando il comportamento del sistema posso capire se sta andando in overfitting o meno.

Ovviamente i modelli complessi funzionano: nel caso del polinomio di grado 9 non andava bene poiché avevamo fornito pochi dati, se forniamo molti più dati diventa perfetto.

Questo suggerisce che più dati abbiamo e meglio è.

- Polinomio con M = 9



I modelli deep-learning sono addestrati su milioni di dati.

## ESEMPIO DI REGRESSIONE

La comunità scientifica ha accumulato una serie di dati relativi a problematiche e le ha resi disponibili alla comunità per usarli come esempi di training.

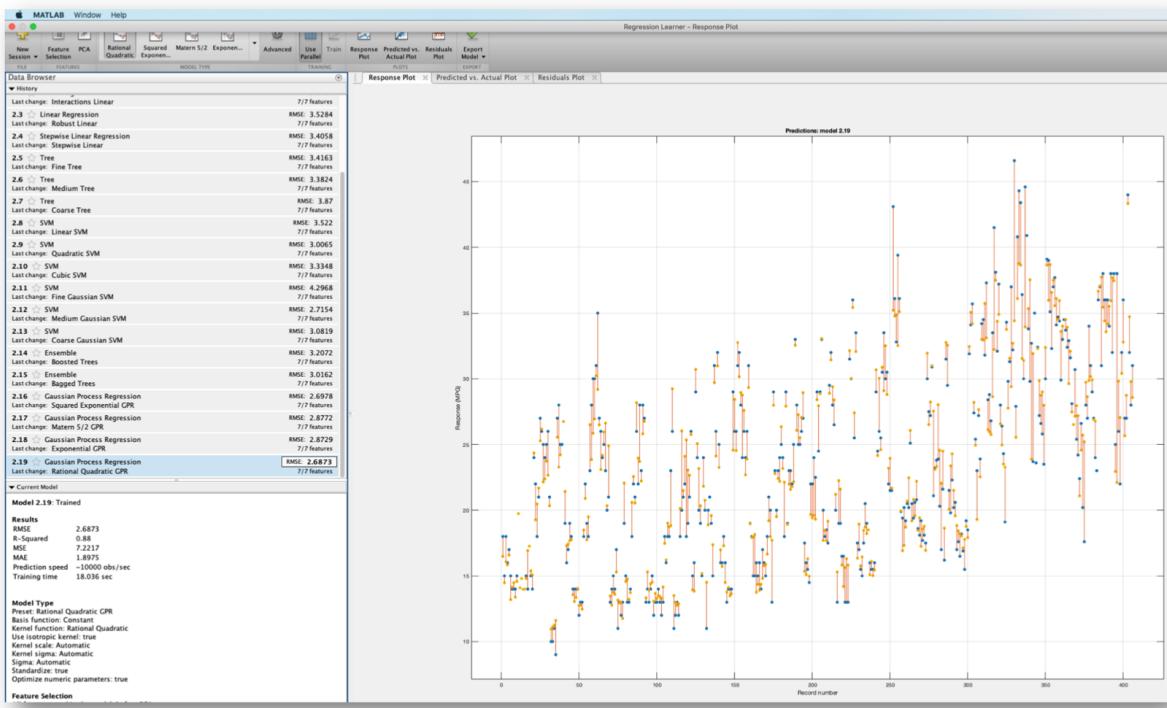
Uno di questi dataset è costituito da un insieme di auto americane con 406 auto descritte da un insieme di parametri (consumi, potenza, anno, peso)... non tutti i parametri sono significativi per risolvere il problema che consiste nel **determinare qual è il consumo dell'auto sulla base di questi input**.

Di queste 406 auto devo organizzare un dataset: una parte per l'addestramento (60%), una per la validazione (20%) e l'altra per il test (20%).

**Che cos'è il dataset concretamente?** Un file con una matrice di numeri: ogni riga è un automobile e le colonne sono i valori in input.

	Acceleration	Cylinders	Displacement	Horsepower	Model_Year	Weight	Origin	MPG	
82	16	8	302	140	72	4294 USA	13		
83	14	8	318	150	72	4077 USA	14		
84	14.5000	4	121	112	72	2933 Sweden	18		
85	18	4	121	76	72	2511 Germany	22		
86	19.5000	4	120	87	72	2931 France	21		
87	17	4	96	69	72	2389 France	26		
88	16	4	122	86	72	2302 USA	22		
89	17	4	97	92	72	2288 Japan	28		
90	14.5000	4	120	97	72	2505 Japan	23		
91	15	4	98	80	72	2164 USA	28		
92	16.5000	4	97	88	72	2100 Japan	27		
93	13	8	350	175	73	4100 USA	13		
94	11.5000	8	304	100	73	3670 USA	14		
95	13	8	350	145	73	3884 USA	13		
96	14.5000	8	302	137	73	4042 USA	14		
97	12.5000	8	318	150	73	3777 USA	15		
98	11.5000	8	429	198	73	4952 USA	12		
99	12	8	400	150	73	4464 USA	13		
100	13	8	351	158	73	4363 USA	13		
101	14.5000	8	318	100	73	4234 USA	14		
102	11	8	440	215	73	4735 USA	13		
103	11	8	455	225	73	4951 USA	12		
104	11	8	360	175	73	3821 USA	13		
105	16.5000	6	225	105	73	3121 USA	18		
106	18	6	250	100	73	3278 USA	16		
107	18	6	232	100	73	2994 USA	18		
108	16.5000	6	250	88	73	3021 USA	18		
109	16	6	198	95	73	2904 USA	23		
110	21	4	97	46	73	1950 Germany	26		
111	14	8	400	150	73	4997 USA	11		
112	12.5000	8	400	167	73	4906 USA	12		
113	13	8	360	170	73	4654 USA	13		
114	12.5000	8	330	160	73	4495 USA	12		
115	15	6	232	100	73	2789 USA	18		
116	19	4	97	88	73	2279 Japan	20		
117	19.5000	4	140	72	73	2401 USA	21		
118	16.5000	4	108	94	73	2379 Japan	22		
119	13.5000	3	70	90	73	2124 Japan	18		
120	18.5000	4	122	85	73	2310 USA	19		
121	14	6	157	107	73	2472 USA	21		
122	15.5000	4	98	90	73	2265 Italy	26		
123	13	8	350	145	73	4082 USA	15		
124	9.5000	8	400	230	73	4278 USA	16		
125	19.5000	4	68	49	73	1867 Italy	29		
126	15.5000	4	116	75	73	2158 Germany	24		
127	17	4	114	91	73	2343 Germany	20		
128	15.5000	4	121	112	73	2868 Sweden	19		
129	11	8	318	150	73	3399 USA	15		
130	14	4	121	110	73	2660 Sweden	24		
131	13.5000	6	156	122	73	2807 Japan	20		
132	11	8	350	180	73	3664 USA	11		
133	16.5000	6	198	93	74	3102 USA	20		
134	17	6	200	NAN	74	2875 USA	21		
135	16	6	232	100	74	2901 USA	19		
136	17	6	250	100	74	3336 USA	15		

Il problema è molto complesso poiché abbiamo diverse variabili di input.

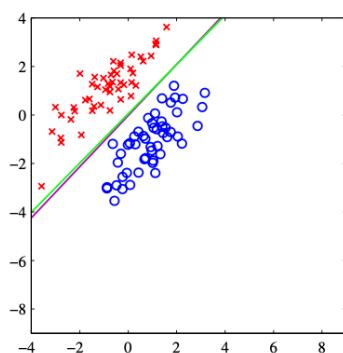


## CLASSIFICAZIONE

Ci sono cose che cambiano: funzione di errore ecc ma concettualmente i passi sono gli stessi.

Un problema di classificazione è modellata come una probabilità: i modelli predicono per ciascun output un valore compreso tra 0 e 1 che rappresenta la probabilità di quella classe dato un input.

Nella classificazione si prendono i dati e determiniamo un confine di decisione che separa le varie classi. Supponiamo di avere un problema relativo alla classificazione spam-mail. Se vado a plottare, le croci potrebbero essere spam e pallini le mail. Dal punto di vista geometrico, dobbiamo trovare la retta che separa le due classi, una volta determinata, se arriva una nuova mail posso dire se questa è spam o meno.

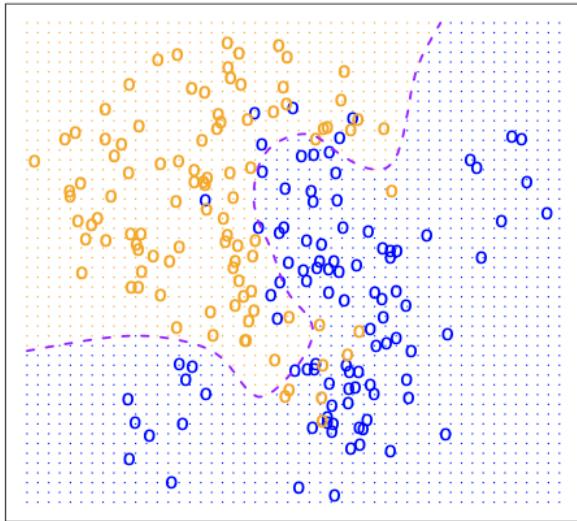


Questo è un problema semplice, linearmente separabile.

---

## CLASSIFICAZIONE MULTI CLASSE

Il problema è quando le classi non sono linearmente separabili.



---

## ESEMPIO DI CLASSIFICAZIONE

Uno dei dataset è quello dei fiori *Iris*: è un dataset messo su per illustrare diversi metodi statistici.

Ci sono 150 fiori Iris che sono di 3 tipologie diverse: versicolor, setosa e virginica. Ciascuno di questi fiori ha 4 input: lunghezza e larghezza del petalo, lunghezza e larghezza del sepalo.

Dati questi 150 fiori, l'obiettivo è classificare i diversi Iris.

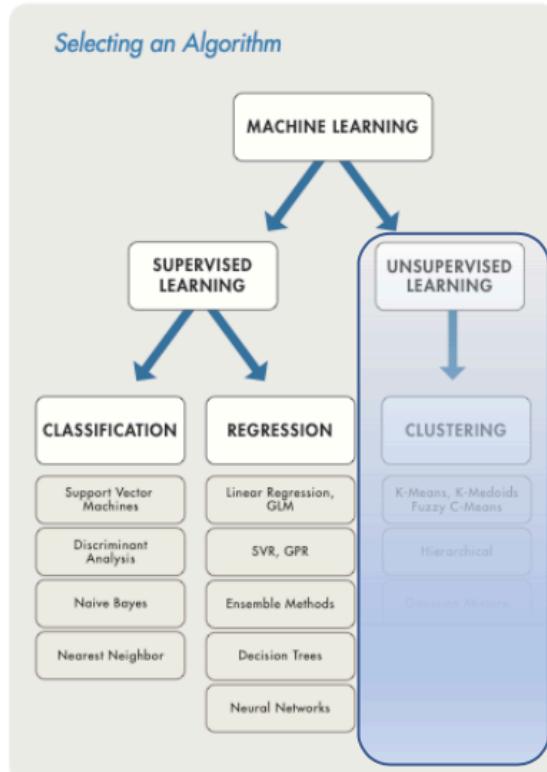
Per capire l'accuratezza della classificazione si usa una *matrice di confusione* che ha tante righe e colonne quante sono le classi da predire, nel caso di iris ho 3 classi e quindi una matrice  $3 \times 3$ : sulle righe indico il valore dell'output corretto da predire, sulle colonne ho i dati predetti dal modello. Sulla diagonale ho le risposte corrette e per avere la percentuale basta sommare i valori della diagonali e dividerli per il numero di dati in input (ad esempio 150 per iris). La matrice di confusione è calcolata sul **test-set**, ovviamente.

---

## METODI ML SUPERVISIONATI

Il ML è diviso in 2 paradigmi principali:

- Metodi supervisionati (classificazione e regressione).
- Metodi non supervisionati.



Le tecniche di classificazione e regressione sono molte.

## DEFINIZIONE GENERALE DI APPRENDIMENTO

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

---

## RICONOSCIMENTO DI CARATTERI SCRITTI A MANO

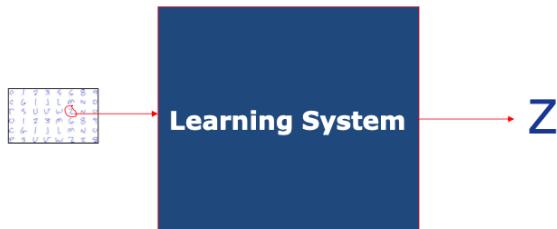
In questo caso il problema si può formulare come un problema di classificazione: ho diversi caratteri che devono essere riconosciuti.

Se ho dei numeri ho un problemi di 10 classi (da 0 a 9).

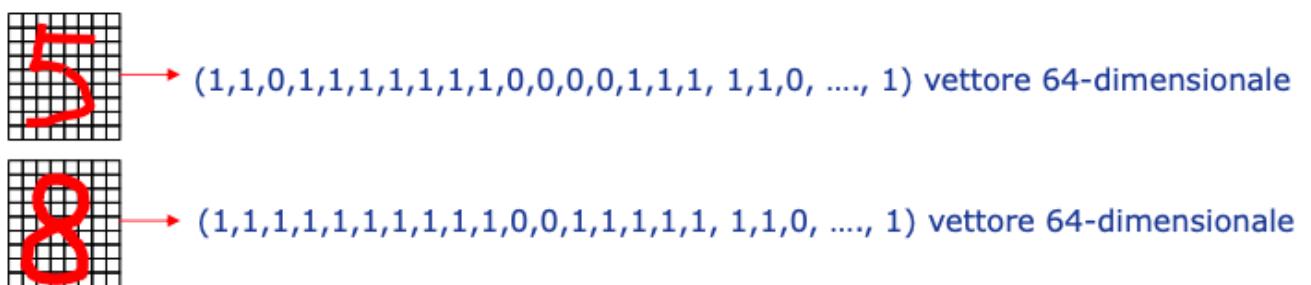
- Problema del riconoscimento dei caratteri scritti a mano
    - **Compito T:** riconoscere i caratteri scritti a mano
    - **Misura delle prestazioni P:** percentuale di caratteri classificati correttamente
    - **Esperienza E di addestramento:** una base di dati di caratteri scritti a mano con le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

## PROGETTARE UN SISTEMA DI PARRENDIMENTO

- Passo iniziale: **trattiamo il sistema di apprendimento come una scatola nera (black box).**  
Data un'immagine di un carattere l'obiettivo è che il sistema predica qual è il carattere corretto.

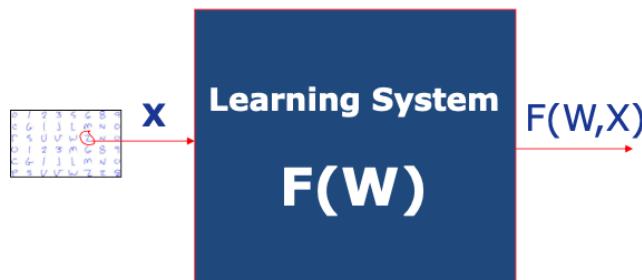


- Secondo passo: **devo avere un training set** con un certo numero di caratteri con la relativa classe di appartenenza. Più ne abbiamo, meglio è.
- Passo 3: **rappresentare l'esperienza.**



Per ciascun pixel indico se è bianco o nero ottenendo un vettore che darò in input.

- Passo 4: **scegliere una rappresentazione per la black-box**, ad esempio una rete neurale.  
È necessario scegliere una funzione  $F$  per approssimare la black-box. Per un input  $X$ , il valore di  $F$  fornirà la classificazione di  $X$ .



- Passo 5: **apprendere/modificare i pesi** usando un algoritmo. Nel caso delle reti neurali l'algoritmo è un gradiente discendente.

Devo calcolare la funzione di errore data come la differenza tra risposta corretta e label fornita dal modello. Modificando i pesi, a una successiva iterazione, il valore dell'errore diminuisce iterativamente fino al tendere di zero.

L'apprendimento finisce o quando il training-set è terminato o quando si fissa una soglia del valore di errore.

- Passo 6: **test del sistema** → dopo aver completato il learning fornisco dati esclusivi per vedere se il sistema ha appreso correttamente.

### SPAZIO DELLE FEATURE

I nostri *dataset* rappresentano diversi domini applicativi e ogni entità ha un insieme di variabili di input.

Queste variabili di input sono dette **features** e a prescindere dal problema possiamo rappresentare gli oggetti del mondo reale con dei vettori di features, la cui lunghezza dipende dal numero di parametri.

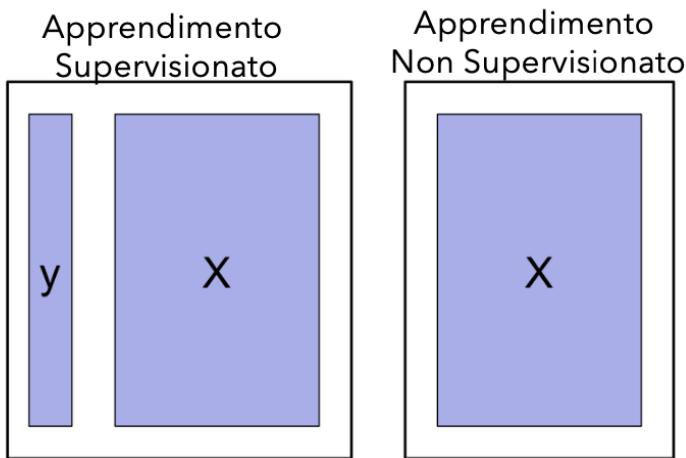
## Lez 4 → ML APPRENDIMENTO NON SUPERVISIONATO

Un altro paradigma di apprendimento è quello **non supervisionato**.

Qual è la differenza?

Quando lavoriamo col ML usiamo un dataset ben definito. Anche in questo caso lavoriamo con dati con la **differenza** che abbiamo un insieme di campioni (pazienti, immagini) descritti da un insieme di **features** ma non abbiamo una variabile di output da predire (**y**).

Le *label* quindi non ci sono.



### QUAL È L’OBIETTIVO?

**L’obiettivo** è scoprire cose interessanti sulle misurazioni delle variabili di input.

Lo facciamo con procedure che sulla base delle *features* sono in grado di trovare delle similitudini.

Tutte le analisi di mercato dove si cerca di capire i gusti dei clienti sfruttano metodi non supervisionati perché sulla base di info quali età, abitudini, trovano gruppi di persone che condividono dei gusti e indirizzano verso loro specifici riferimenti.

Le tecniche non supervisionate si occupano anche di altro ma una gamma di applicazioni riguarda le tecniche di visualizzazione: quando abbiamo un dataset che rappresenta entità con features, se sono poche posso visualizzarli nel piano cartesiano per fare delle considerazioni e avere un’idea della natura di essi, **MA nel momento in cui lavoriamo con dati con features  $\geq 3$  come faccio a visualizzarli?**

**La visualizzazione è fondamentale, esistono quindi delle tecniche come la PCA che consente di trasformare i nostri dati dallo spazio originario in 3D preservando i dati e visualizzandoli.**

Uno degli ambiti delle tecniche non visionate è quindi la visualizzazione dei dati.

L'altro è individuare delle similitudini che danno info nascoste all'interno dei dati a disposizione. Questo compito prende il nome di **clustering**.

Nell'ambito dell'ELIM il clustering serve per la segmentazione.

---

## LE SFIDE DELL'APPRENDIMENTO NON SUPERVISIONATO

L'apprendimento non supervisionato è più complesso poiché non abbiamo riferimenti per guidare l'apprendimento.

Quando abbiamo un output non sapremo mai se effettivamente avremo un risultato corretto, bisogna affidarsi a esperti del dominio applicativo considerato.

- In ogni caso, le tecniche non supervisionate giocano un ruolo fondamentale
  - Esempio
    - Sottogruppi di pazienti affette da cancro al seno raggruppate sulla base delle rispettive espressioni geniche
    - Gruppi di acquirenti caratterizzati dalle storie di acquisto e visualizzazione di prodotti
    - Film raggruppati sulla base dei giudizi assegnati da persone che lo hanno già visto

---

## CLUSTERING

Dato un dataset di info non etichettate come viene analizzato?

Il clustering è un algoritmo che deve individuare un insieme di gruppi chiamati *cluster* all'interno del nostro insieme di dati con l'obiettivo di partizionare i dati, e tutte le osservazioni devono essere simili a quelli dello stesso cluster e diverse da quelli fuori dal gruppo clusterizzato.

È fondamentale stabilire in che modo due osservazioni sono simili.

Qual è la misura della similitudine? Lo si fa introducendo delle metriche, ad esempio quella della *distanza euclidea* che però non è efficiente per tutti i tipi di dati e quindi ci sono diverse tecniche.

La cosa fondamentale è che la conoscenza del dominio applicativo deve essere elevato, anche per stabilire quali metriche adottare.

---

## TECNICHE DI CLUSTERING

Ci sono molte tecniche di clustering. Le più importanti sono 2:

**K-MEANS:** è un algoritmo di clustering dove l'obiettivo è cercare di individuare un certo numero di raggruppamenti. La caratteristica è che devo dare in input quanti cluster voglio individuare ( $k$  sono le variabili), anche se non sappiamo effettivamente quanti cluster possiamo individuare in generale.

**CLUSTERING GERARCHICO:** non è necessario dare in input il numero di cluster, è l'algoritmo che li individua e sceglie il numero più conveniente. Costruisce un albero di clusterizzazione. Ci sono 2 tipi di clustering gerarchico:

- Divisivo.
  - Agglomerativo.
- 

## CLUSTERING K-MEANS

Ipotizziamo di avere un dataset con 150 osservazioni (campioni) in uno spazio 2D: i parametri con cui descrivo queste info sono 2 (ad esempio 150 studenti con 2 features come altezza e peso).

**K-means** chiede quanti cluster vogliamo trovare, fatto ciò, l'algoritmo si ritrova i raggruppamenti.

## DETTAGLI DELL'ALGORITMO

Supponiamo che i cluster da individuare sono  $k$  e li chiamiamo  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K$ . Ciascun cluster  $\mathcal{C}_i$  contiene gli indici corrispondenti alle osservazioni che ricadono in quel cluster.

### Le proprietà dell'algoritmo:

Se faccio l'unione degli insiemi dà come risultato tutto il dataset. Ogni campione appartiene almeno a un cluster.

Se prendiamo una coppia di cluster e facciamo l'intersezione è vuota: un dato può appartenere a 1 e un solo cluster.

C'è una variante che elimina la seconda proprietà.

---

## COME DEVE FUNZIONARE

Premessi i vincoli citati, nel momento in cui un gruppo è in un cluster, devono essere simili tutti gli appartenenti.

## MISURA DI VARIABILITÀ'

La **variabilità intra-cluster** è la misura di variabilità degli elementi all'interno di un cluster, detta anche VIC (quanti sono diversi gli elementi del cluster  $C_k$ ).

**Preso ciascun cluster dei K devo minimizzare la somma di VIC: devo dividere in K cluster in modo che la intra-cluster sia più piccola possibile.**

## METRICHE

Come stabiliamo se siamo vicino o lontani? Il quadrato della distanza euclidea. Su tutti i cluster devo minimizzare la distanza euclidea dei cluster che appartengono. Questa funzione prende il nome di *funzione obiettivo k-means*. per risolvere la funzione si applica un algoritmo iterativo che a ogni passo trova il partizionamento che si muove al valore minore, finché si arriva a un passo in cui i valori non sono tendenti al minimo.

---

## FUNZIONAMENTO

Inizialmente abbiamo il nostro dataset e vogliamo raggrupparli in K cluster. Assegno i dati casualmente ai K cluster, fatto ciò, itero finché non ci saranno ulteriori variazioni:

Per ognuno dei K cluster trovati al primo passo, calcolo il centroide (media). Lo faccio per ogni cluster.

Quando ho i  $K$  centroidi, riassegno di nuovo i dati ai cluster in maniera tale che ognuno venga assegnato al cluster il cui centroide è più vicino.

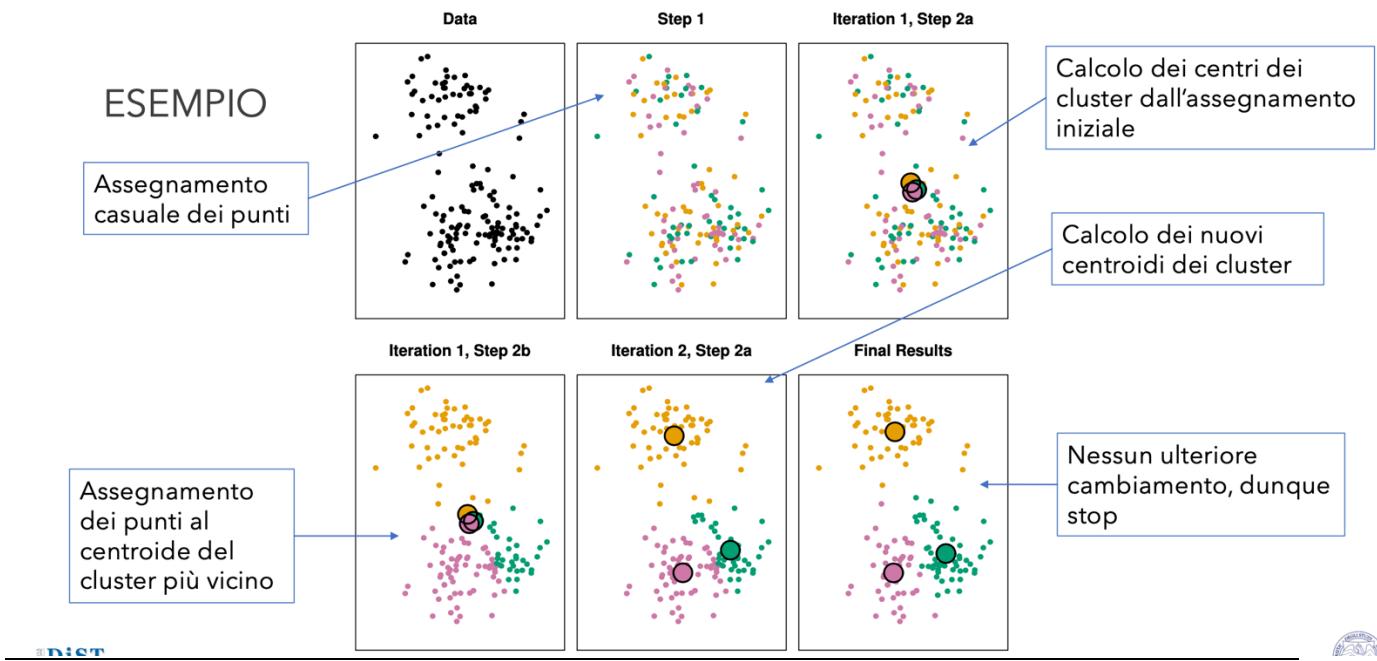
Il nuovo centroide è cambiato: ricalcolo di nuovo il centroide e vado avanti in questo modo. L'algoritmo converge fino a un certo punto in cui i nuovi centroidi rispetti ai precedenti non variano e di conseguenza non conviene più continuare e ho trovato i k cluster.

---

## PROPRIETA' DELL'ALGORITMO

Se ho una funzione obiettivo convessa, allora converge ed è anche il caso migliore ma di solito la funzione obiettivo non è convessa e non fornisce il minimo globale. Se l'assegnamento casuale ci pone in un punto sfavorevole,

potrei non arrivare al minimo globale, quindi si riproduce l'algoritmo più volte sugli stessi dati, finché non trovo il minimo più conveniente.

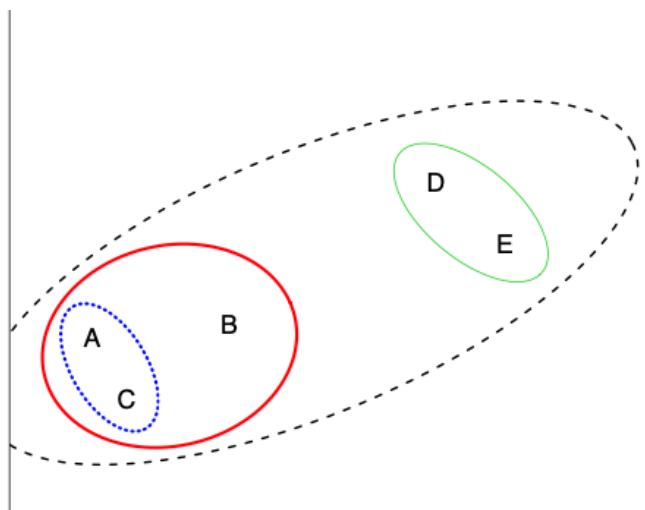


## CLUSTERING GERARCHICO

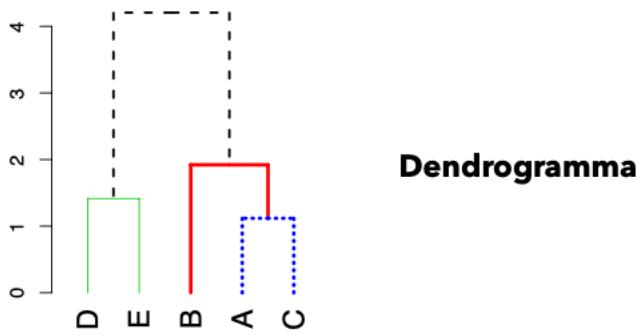
Non dobbiamo specificare il numero di cluster di partenza, al contrario del k-means.

Studiamo l'agglomerativo (bottom-up) dove considero tutti i dati come un singolo cluster e poi li unisco in base alle similitudini.

L'algoritmo gerarchico parte considerando ogni foglia come un singolo cluster e poi accomuna i singoli simili in un cluster più grande, fino ad arrivare a un unico cluster.



L'algoritmo termina quando riesco a creare un singolo cluster.



Se parto da basso nel **dendrogramma** avrò prima gli elementi vicini. Questo algoritmo è usato molto per fare l'analisi delle espressioni bioinformatiche.

### ESEMPIO

45 dati in uno spazio 2D. ci sono 3 classi distinte e un possibile dendrogramma è quello in figura.

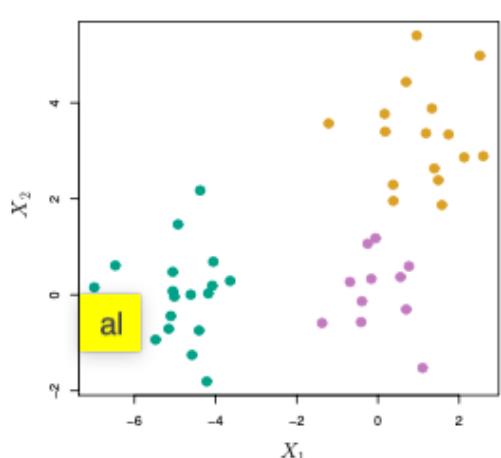
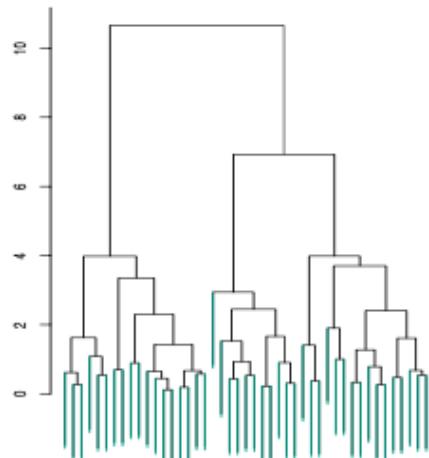


Figura - 2000/2004 - Data 1 - Dati A - Clusters



### SCELTA DEI CLUSTER

Dopo che abbiamo il dendrogramma devo tagliarlo: sulla base del **cut** avrò un numero di clusterizzazioni.

### COME DEFINIAMO LA DISSIMILARITA' O SIMILARITA'?

Per stabilire il grado di similarità devo introdurre una metrica, anche qui potremmo usare l'eucleidea, anche se non potrebbe essere efficiente.

Ho anche bisogno di stabilire un criterio in base al quale presi due cluster con almeno un'osservazione posso prenderli e fonderli, ci sono 4 possibilità:

**Complete linkage:** prende due cluster e li fonda se tra questi 2 cluster, rispetto a tutti gli altri, la più grande distanza tra le osservazioni è la più piccola di tutte.

**Single Linkage:** prendo la distanza più piccola tra le osservazioni  
**Average linkage:** distanza media tra le osservazioni

**Centroide:** distanza tra i centroidi delle osservazioni

A seconda dei criteri, i risultati cambiano, con l'average abbiamo dendrogrammi più bilanciati mentre nel caso del single c'è una maggiore variabilità.

Il migliore non sappiamo qual è, dipende dai casi considerati e dal dataset.

---

### DISTANZA EUCLIDEA: PROBLEMA DELLA MISURA

Non è sempre buona perché potrei perdermi delle clusterizzazioni ottimali, si usa spesso una misura chiamata **correlazione** che specifica la correlazione tra le features e le accomuna, così facendo ho dei cluster con caratteristiche comuni ben definite.

---

### CONSIDERAZIONI PRATICHE PER IL CLUSTERING

I dati vengono prima pre processati prima di essere dati in pasto agli algoritmi.

In caso di clustering gerarchico:

- Scegliere la misura di dissimilarità
- Linkage da usare
- Dove tagliare il dendrogramma

In caso di k-means:

- Quanto deve valere  $K$ ?

## Lez 5 → FORMAZIONE DELLE IMMAGINI

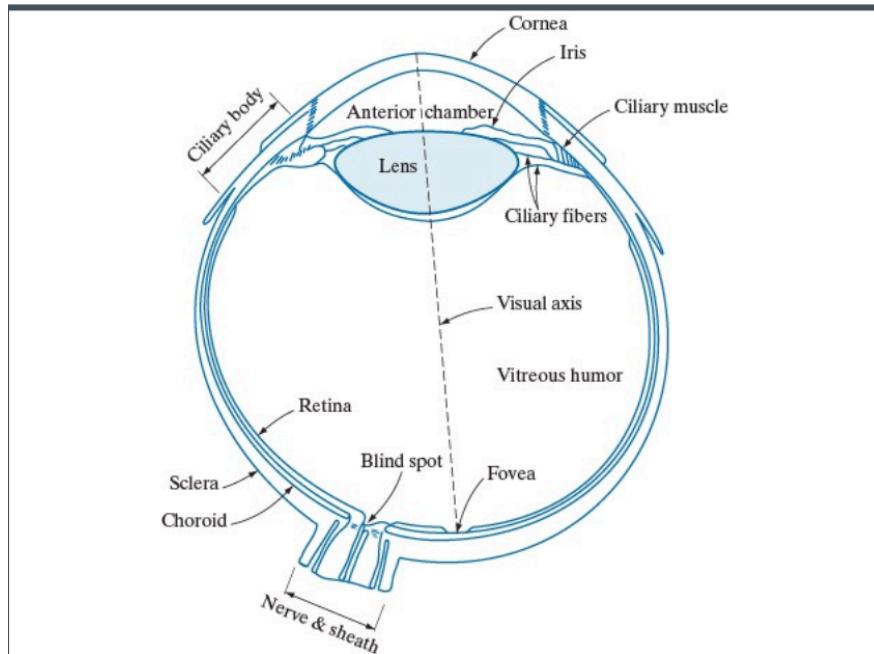
Dobbiamo capire innanzitutto come sono formate le immagini mediante l'uso di macchine che catturano immagini.

Quali sono gli elementi che caratterizzano un'immagine 2D acquisita con un dispositivo?

Prima di fare ciò capiamo com'è fatto il sistema visivo dell'essere umano.

### STRUTTURA DELL'OCCHIO UMANO

L'occhio ha un diametro di 20mm ed è racchiuso da 3 membrane esterne. La *cornea* è la membrana più esterna dell'occhio, abbiamo poi un tessuto trasparente che protegge l'occhio e la parte posteriore è la *sclera*. Al di sotto di queste membrane c'è la *coroide*. La coroide è una rete di vasi sanguigni che rappresenta la fonte di nutrimento dell'occhio.



Per quanto riguarda la coroide, la parte anteriore è costituita dall'*iride* che è la componente che può contrarsi o espandersi per regolare la quantità di luce da "assorbire". All'interno ci sono le *lenti* che sono costituite da mix di acqua e proteine che assorbono l'8% della luce che passa attraverso la pupilla e lo scopo di esse è quella di raccogliere la luce e la messa a fuoco.

La parte posteriore è rappresentata dalla *retina*: possiamo immaginarla come la pellicola della macchina fotografica o come il dispositivo di imaging sul quale si imprime l'immagine: la luce passa attraverso e viene impressa sulla retina. La cosa interessante è che siamo in grado di distinguere una scena perché sulla retina ci sono 2 tipologie di recettori per vedere sia di giorno che di notte: i **coni** e i **bastoncelli**.

I **coni** sono minori rispetto ai bastoncelli e sono localizzati nella parte centrale della retina, chiamato ***fovea***. I coni sono sensibili ai colori e ciascun cono è connesso a un'estremità nervosa. I coni sono attivi di giorno e consentono la versione fotonica. I **bastoncelli** sono circa 100 milioni, sono poco sensibili ai colori e permettono la visione notturna, anche perché sono meno adatti al ricevimento dei “dettagli”, infatti condividono la stessa cellula nervosa.

---

## FORMAZIONE DELLE IMMAGINI NELL'OCCHIO UMANO

C’è una sorgente luminosa, di solito il **sole**, che colpisce oggetti in una scena, questi oggetti assorbono la luce ricevuta e la riflettono e la luce riflessa è acquisita dai dispositivi, in questo caso dai nostri occhi. Si vanno poi ad imprimere sulla retina.

La distanza tra il centro del cristallino e la retina è fissa (17mm) ed è detta ***punto focale***. In questo modo abbiamo la possibilità di mettere a fuoco perché c’è la possibilità di variare la lunghezza focale. Nell’occhio umano avviene grazie ai muscoli del corpo ciliare che modellano il cristallino e fanno passare più o meno luce.

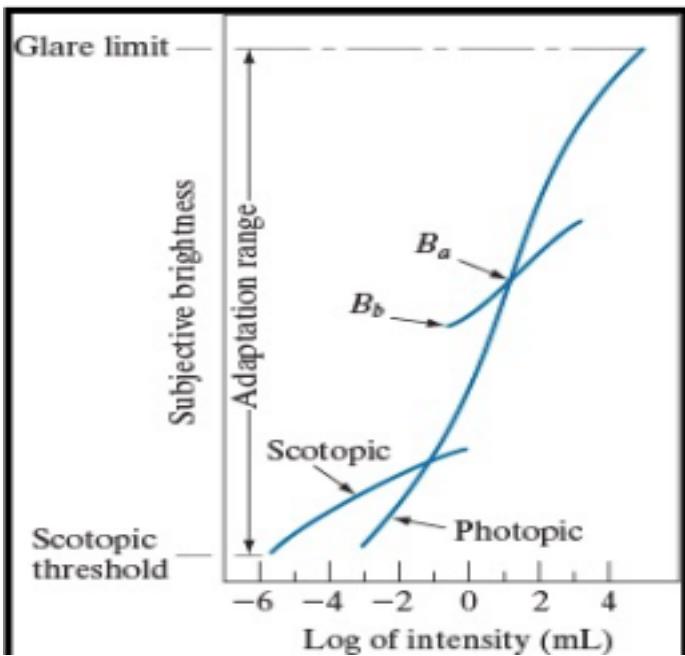
## LUMINOSITA’ PERCEPITA

Per quanto riguarda l’intensità di luce percepibile dall’occhio umano, è molto elevata. I diversi valori di intensità sono un’enorme quantità ma quello che il

nostro occhio percepisce sono circa  $10^{10}$ , un valore estremamente elevato. Questo intervallo non possiamo percepirllo simultaneamente ma solo un sottoinsieme di livelli vicini alla luminosità della scena possiamo percepire. Questo range si adatta rispetto all’intensità di luce dell’ambiente: questo è motivo per il quale se entriamo in una stanza buia non vediamo nulla e poi usciamo con una luce fortissima non vediamo bene, dobbiamo prima focalizzare il tutto [...].

Ovviamente l’intensità di luce percepita è soggettiva.

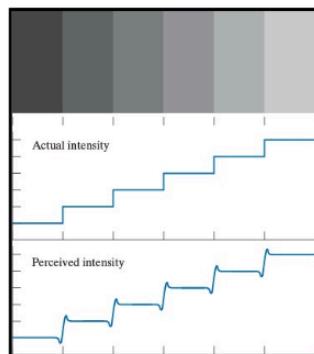
La luminosità percepita dipende dalla luminosità ambientale, *B<sub>a</sub>*. Al di sotto la soglia *B<sub>b</sub>* di intensità di luce, è percepito il nero. Il limite superiore è meno importante poiché una luce più brillante può traslare *B<sub>a</sub>*.



L'intensità che noi percepiamo non è solo funzione dell'intensità media dell'ambiente in cui ci troviamo ma è condizionata anche da fattori di contesto e anche dal contrasto e altri fattori.

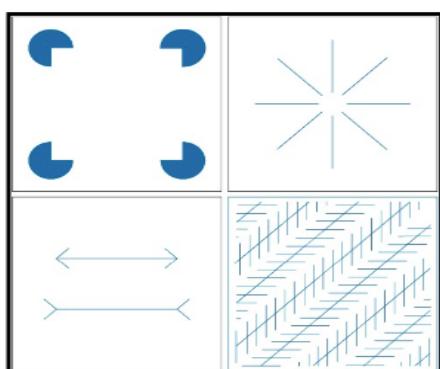
#### INTENSITÀ PERCEPITA

- La luminosità percepita non dipende solo dall'intensità
- Ad esempio, le bande di Mach sono regioni di intensità costante
- Tuttavia, il confine tra due bande appare più scuro nella regione più scura e più chiaro dal lato chiaro della banda



#### ILLUSIONI OTTICHE

Il nostro sistema è sensibile ad alcune situazioni e percepisce oggetti che non corrispondono alle proprietà reali.



Questo per quanto riguarda la visione umana.

## IMMAGINI DIGITALI

Abbiamo detto che le immagini digitali sono ottenute da un dispositivo di imaging. Sono dette digitali poiché la luce che il dispositivo acquisisce è un segnale analogico che è convertito in un segnale elettrico e poi viene discretizzato per ottenere l'immagine digitale.

## OTTICA DI BASE

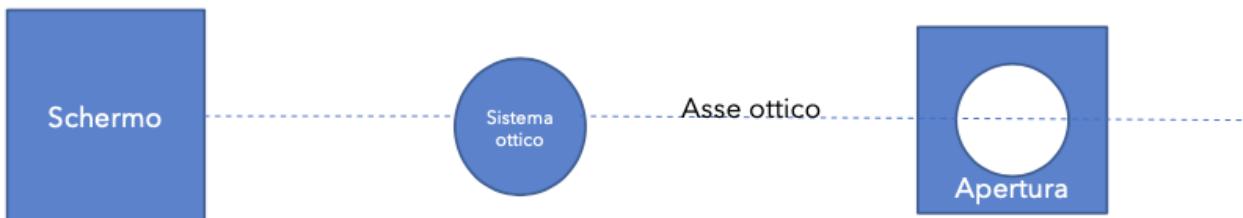
Quando osserviamo un'immagine digitale, la scena ripresa è soggetta a delle distorsioni per cui bisogna cercare di capire qual è la relazione che esiste tra il mondo reale 3D e quello rappresentato in 2D sulla scena. Vogliamo capire come tutto ciò influenza l'immagine digitale sia in termini geometrici che dall'intensità del valore dei pixel e dalla fotometria (processo di assorbimento della luce che riflette l'immagine della scena).

Possiamo vedere un dispositivo di imaging come composto da più componenti:

Lente

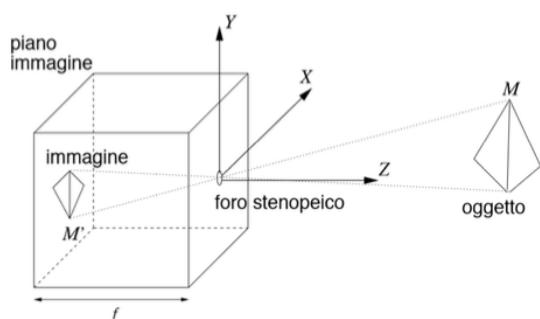
Schermo: elemento su cui viene generata l'immagine

Elemento che regola la luce che acquisisce il dispositivo



## GEOMETRIA DELLA FORMAZIONE DELL'IMMAGINE

Per modellare le caratteristiche della proiezione geometrica, dato che passiamo da 3D al 2D, il modello più semplice da considerare è la **telecamera pinhole**.



La telecamera *pinhole* è una scatola dove davanti abbiamo un piccolo foro, chiamato *foro stenopeico* o *pinhole*, e all'interno abbiamo una parte della

scatola che è il piano immagine. Supponiamo un oggetto del mondo 3D, fissiamo un sistema di coordinate che è centrato col foro stenopeico, l'asse Z ci fornisce la distanza dal foro all'oggetto osservato.

La dimensione del foro è importante poiché quando c'è un oggetto del mondo 3D, ad esempio un punto  $M$ , da questo punto arriva la sorgente luminosa che colpisce l'oggetto e la luce viene riflessa. La quantità di luce determina l'aspetto di quello che si forma nella telecamera: più luce entra più proiettiamo su un singolo punto un insieme di raggi. Più grande è il foro più l'immagine è sfocata ma più chiara, e viceversa: questo modello ovviamente non è efficiente poiché il foro deve essere piccolo ma l'immagine è scura, se usiamo delle lenti invece riusciamo a convogliare più luce ma su un solo punto, avendo quindi un'immagine perfetta.

Ci interessa stabilire la relazione tra le coordinate 3D della scena e quelle 2D del piano immagine.

vogliamo quindi sapere la relazione tra le coordinate 3D  $X Y Z$  e le coordinate 2D  $X' Y' Z'$  dove  $Z'$  non ci interesserà poiché non esiste...

le relazioni possono essere trovate in due modi: o con la relazione di similarità dei triangoli o osservando che i punti  $P, O, P'$  sono collineare da cui derivo che la coordinata

$$X' = \lambda_X, Y' = \lambda_Y \text{ e } Z' = -\lambda_Z$$

Z viene chiamata **effetto scorcio** poiché osserviamo che più la scena è lontana, più Z è grande e quindi sul piano immagine la scena è più piccola e l'immagine che si riproduce sul piano immagine è invertita, sia orizzontalmente che verticalmente, ecco perché il segno meno.

Queste relazioni definiscono la proiezione prospettica, la sua caratteristica è che le linee parallele appaiono sul piano immagine come convergenti. Questa cosa la possiamo caratterizzare matematicamente poiché: consideriamo una linea che nella nostra scena che attraversa un punto  $x_0 y_0 z_0$ , supponiamo che la direzione sia  $u, v, w$  tutti i punti della retta passante per  $x_0 y_0 z_0$  che si trovano nella direzione  $u, v, w$  li posso descrivere come  $x_0 + \lambda_U, y_0 + \lambda_V, z_0 + \lambda_W$ . Se prendo un punto su questa retta e lo vado a proiettare su un piano immagine 2D avrò che la coordinata  $X$  la posso esprimere come (vedi formula. minuto 48 ...

Nella proiezione prospettica se la distanza media del dispositivo di imaging è abbastanza elevata e i punti dell'oggetto sono poco profondi, la proiezione prospettica può essere approssimata con la **proiezione ortografica**, significa dire matematicamente che: se indico con  $Z_0$  la distanza media dei punti

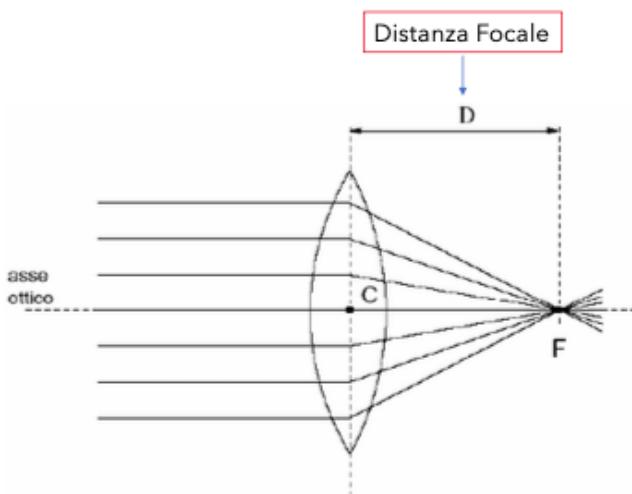
dell'oggetto in questione dalla camera, e dato che la profondità di un oggetto è più o meno sempre la stessa posso approssimarla con una costante  $S$ , quindi posso esprimere le coordinate  $X' Y'$  come  $sX e sY$ .

---

## LENTI SOTTILI

Abbiamo detto che la *pinhole* è la modalità più semplice, ma abbiamo bisogno di immagini di altissima qualità, cosa che il modello di cui sopra non garantisce poiché ci basiamo semplicemente su un foro che riproduce immagini. Ho necessità di convogliare la luce proveniente dalla scena in un singolo punto e uso delle lenti. Nel nostro occhio abbiamo il cristallino che si deforma e fa passare luce e nel dispositivo di imaging abbiamo un sistema di lenti, detto **lente sottile**.

- 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 sono 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



## CERCHI DI CONFUSIONE

Più un soggetto è lontano dal punto di messa a fuoco, più grandi saranno i cerchi che il soggetto forma sul sensore.

Quando un cerchio è piccolo, nella foto risulterà indistinguibile da un punto, questo genera la zona nitida che chiamiamo profondità di campo.

Possiamo spostare la lente verso l'oggetto della scena finché non rispettiamo questa distanza e l'oggetto è a fuoco.

Anche in questo caso si chiama distanza focale, ma è un concetto differente rispetto alla telecamera pinhole.

## FOTOMETRIA DELLA FORMAZIONE DELL'IMMAGINE

In che modo l'oggetto della scena riflette la luce e come il dispositivo assorbe la luce?

La luminosità di un pixel di un'immagine dipende dalla luminosità della parte di superficie della scena che è illuminata dalla sorgente luminosa: se stiamo riprendendo un oggetto illuminato dal sole, la luminosità del pixel dipenderà dalla luminosità della porzione dell'oggetto.

Dipende anche dalla luminosità del raggio proveniente dalla sorgente luminosa e anche dalle capacità di riflessione dell'oggetto.

Semplificando possiamo dire che la luminosità di un pixel dipende dalla luminosità della scena.

Quando osserviamo con i nostri occhi delle scene ci sono delle ambiguità: esse sono dovute dal fatto che la luminosità dipende da molti fattori: oltre alla luminosità della sorgente luminosa anche dell'intero ambiente.

Il nostro occhio è bravo ad individuare le ambiguità però ci sono situazioni con maggiore ambiguità e anche l'occhio umano potrebbe non interpretare bene la scena.

Tutto ciò per dire che la luminosità dipende da  $n$  fattori.

Per mostrare in che modo viene rappresentato il valore di intensità di un pixel possiamo assumere 2 misure di base:

Colore: non è una proprietà degli oggetti ma della luce. Bisogna capire in che modo un oggetto riflette la luce che riceve dalla sorgente luminosa, prendiamo ad esempio il sole che è una sorgente luminosa *puntuale*.

Intensità.

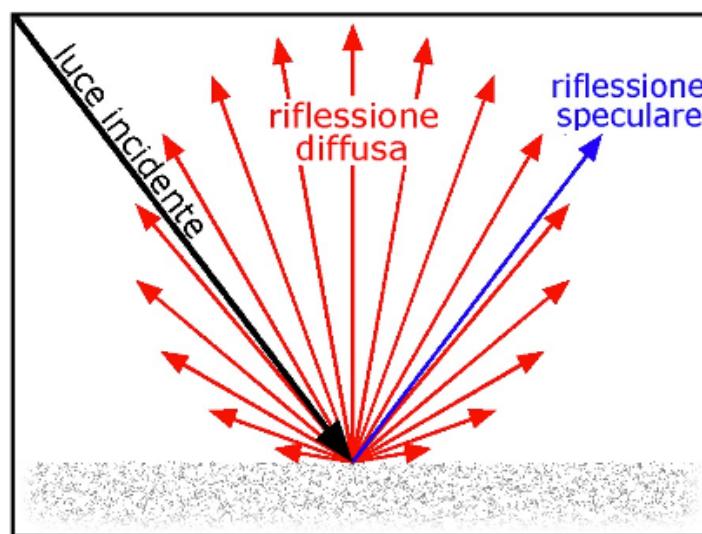
Ci sono due modelli di riferimento per capire come si riflette la luce:

### MODELLO DI RIFLESSIONE DIFFUSA

Tutta la luce che arriva all'oggetto è riflessa su tutte le direzioni. La luce riflessa non dipende dall'angolo in cui vediamo l'oggetto. Questo dipende dal tipo di superficie di cui è fatto l'oggetto (pietra calcarea...).

L'altro tipo di riflessione è quella SPECULARE (specchio): la luce viene riflessa in una certa direzione, dipendente dall'angolo visuale.

Lo specchio è il modello di riflessione ideale ma in realtà in natura ci sono diverse superfici in cui non c'è una perfetta riflessione *speculare* (*un singolo raggio che colpisce lo specchio ed esce con lo stesso angolo da cui proveniva*) ma ci sono delle *specularità*: si riflette in un'area che nelle immagini.



C'è una relazione tra la luminosità della sorgente luminosa e quello che viene riflesso, data dalla legge di *Lambert* che afferma: l'energia dipende da un fattore chiamato ***albedo*** che è un coefficiente che determina il grado di riflettenza moltiplicata l'intensità della sorgente luminosa per l'angolo da cui arriva.

$$L(x) = \rho(x)E(x)$$

## Lez 6 → ACQUISIZIONE E RAPPRESENTAZIONE DELLE IMMAGINI [capitolo 2 del libro]

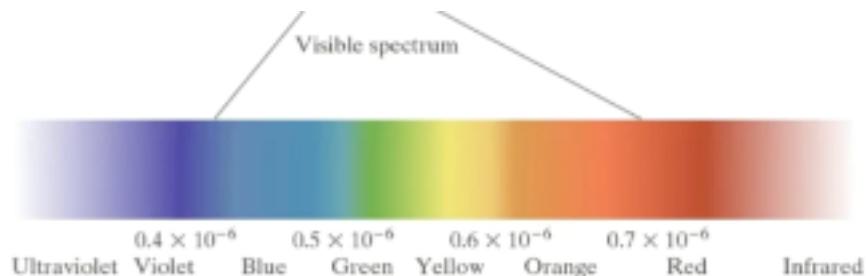
Partiremo con la modalità di acquisizione, cioè che tipo di dispositivi di imaging usiamo per produrre immagini digitali e come passiamo dal dispositivo di imaging all'immagine digitale vera e propria.

### RADIAZIONI ELETTRONAMNETICHE

I dispositivi sono in grado di acquisire la luce visibile che è quella che i nostri occhi recepiscono perché la luce non è altro che un'onda elettromagnetica che costituisce una piccola parte dello spettro elettromagnetico. Le altre onde fuori dallo spettro visibile, l'occhio umano non può recepirle e quindi si usano altri dispositivi che lavorano specificamente in quelle bande di frequenza.

La luce non è bianca, l'esperimento famoso fu fatto da Newton che usò un prisma trasparente e osservò che quando veniva colpito da un raggio di luce esso emetteva un fascio di luce con un range che va dal viola al rosso.

Non c'è una separazione tra le bande: esse degradano in maniera *smooth* nell'altra, sovrapponendosi.



Un'onda elettromagnetica possiamo vederla come un segnale **sinusoidale** che si propaga nello spazio alla velocità della luce oppure possiamo vederlo come un **fascio di particelle (fotoni)** che viaggiano sinusoidalmente alla velocità della luce.

Per questo motivo, essendo sinusoidale è caratterizzata da una lunghezza d'onda che indichiamo con  $\lambda$  e da una certa frequenza.

La relazione tra lunghezza d'onda e frequenza è espressa dalla seguente relazione: la lunghezza d'onda  $\lambda$  è uguale al rapporto della velocità della luce  $c$  e la frequenza  $v$ .

$$\lambda = \frac{c}{v}$$

Come osserviamo, più la lunghezza d'onda è ampia, minore è la frequenza, e viceversa.

Essendo un'onda elettromagnetica un fascio di particelle, è interessante stabilire anche il rapporto che esiste tra le onde di una certa banda e il quantitativo di energia che trasporta, stabilita dalla seguente relazione:  $E = h * \nu$  che afferma che *l'energia è proporzionale alla frequenza*.

$H$  è una costante.

Questa relazione dice che più abbiamo onde ad elevata frequenza e più energia trasportano. Ecco perché i raggi  $X$  e  $\gamma$  sono nocivi per l'uomo.

## RADIAZIONI ELETTRONAGNETICHE

La luce che è composta di onde provenienti da una banda piccola, è definita **monocromatica** ed è caratterizzata solo dal livello di intensità e non dal colore. Siccome il livello di intensità indica quanto un oggetto possa variare di intensità dal colore nero al bianco, la luce **acromatica** è anche detta a livello di grigio.

Se ho un oggetto, di per sé non è visibile ma lo vediamo solo grazie al fatto che egli riceve luce da una sorgente luminosa e poi la riflette. Quando abbiamo un oggetto, ad esempio verde o rosso, è perché il suo materiale assorbe la luce di tutte le bande di colore ed è in grado di riflettere SOLO la frequenza di quel colore: esempio ho una felpa rossa, la felpa assorbe tutte le frequenze e riflette solo il rosso per avere quel colore.

La luce può essere caratterizzata anche in altri modi:

1.Radianza: quantità di energia che trasporta la sorgente luminosa.

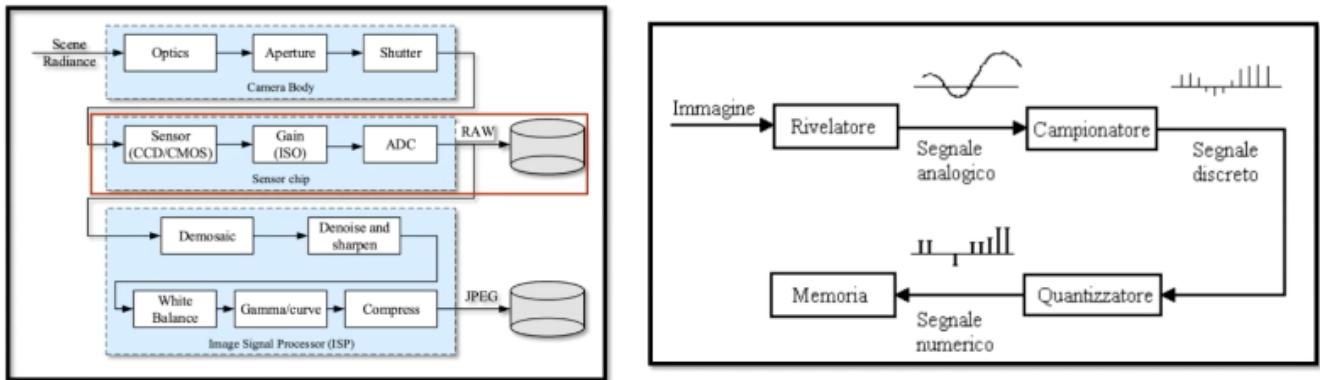
2.Luminanza: si misura in *lume*, è la quantità di energia che possiamo percepire. È importante perché posso avere una sorgente con elevata energia ma con bassa luminanza. Ad esempio una sorgente a infrarossi ha una elevata radianza ma non lo vediamo, lo stesso una sorgente di raggi X.

3.Luminosità: percezione soggettiva della luce.

---

## PROCESSO DI ACQUISIZIONE DELLE IMMAGINI

Vediamo come funziona il processo di acquisizione dell'immagine, dalla luce all'immagine digitale.



Il dispositivo è costituito da un sistema di lenti che hanno lo scopo di convogliare la luce verso un punto specifico del piano immagine, in modo da riprodurre le immagini a fuoco. Quando i raggi passano attraverso l'ottica, c'è un tempo di esposizione dove i fotoni vengono raccolti da sensori. I sensori sono fatti di due tipi: o CCD o CMOS che sono in grado di catturare la quantità di fotoni che lo colpisce e trasforma questa info in un segnale elettrico e producono un'onda che rappresenta un segnale analogico. Successivamente il segnale elettrico passa attraverso degli amplificatori e attraverso un convertitore A/D che rende quel segnale elettrico un segnale digitale, discretizzandolo. Una volta fatto ciò, otteniamo una griglia di pixel dove ognuno di essi ha un valore di intensità che dipende dalla sorgente di illuminazione e dalla scena che riflette la luce.

Quando abbiamo l'immagine digitale, può essere valorizzata in un calcolatore e manipolata.

## IMMAGINI DIGITALI

Un dispositivo di imaging per acquisire immagini nell'ambito della banda visibile è composto da una telecamera, sensori *CCD*, un frame *grabber* (scheda di acquisizione che preleva il segnale analogico del *CCD* e lo discretizza) e una matrice di pixel.

Un *CCD* è una griglia di dispositivi di imaging sensibili ai fotoni che costituisce il piano immagine del dispositivo di imaging. È una matrice  $n \times m$  con molti campi.

Quando la luce passa attraverso l'ottica della camera viene intercettata dalla matrice di sensori e ciascuno di essi raccoglie una quantità di fotoni. Man mano che questi sensori acquisiscono fotoni, scendono in basso dalla matrice finché non arrivano al punto di output, dove c'è l'amplificatore che produce il segnale analogico. La scheda di acquisizione converte il tutto in un'immagine digitale.

in una matrice di punti che ha una risoluzione  $n \times m$ , di solito 512 x 512, e ciascun punto rappresenta l'intensità che va da 0 (nero) a 255(bianco).

La dimensione della matrice del CCD rispetto a quella dell'immagine creata non ha una corrispondenza 1 a 1 ma ci sono relazioni che stabiliscono le coordinate sul piano digitale. Concettualmente può essere utile considerare una corrispondenza 1 a 1.

---

## DISPOSITIVI DI ACQUISIZIONE

Il CCD è considerabile se acquisisco immagini con un dispositivo di imaging che lavora nello spettro visibile.

Esistono però diversi dispositivi per acquisizione, ognuno con peculiarità diverse poiché hanno un sensore sensibile alle diverse bande dello spettro elettromagnetico.

Questi sensori possiamo usarli in maniera diversa per produrre immagini 2D o 3D.

a. Abbiamo un *sensore singolo* che riceve energia dall'onda elettromagnetica che è costituito da un materiale sensibile da quella banda, la risposta del sensore è un segnale elettrico che viene prodotto sulla base della componente elettrica. Sul sensore posso applicare dei *filtri*. Come produco un'immagine digitale con un solo sensore? Posso usare un dispositivo di imaging con un sensore montato su un'asse scorrevole.

b. Potrei avere un array di sensori. Molte immagini aeree montano sensori di questo tipo che sono posizionati in maniera ortogonali rispetto al movimento del veicolo. Da un lato l'array crea una dimensione e dall'altro il movimento dell'aereo crea l'altra dimensione.

c. Potrei avere sensori disposti su una griglia, come CCD.

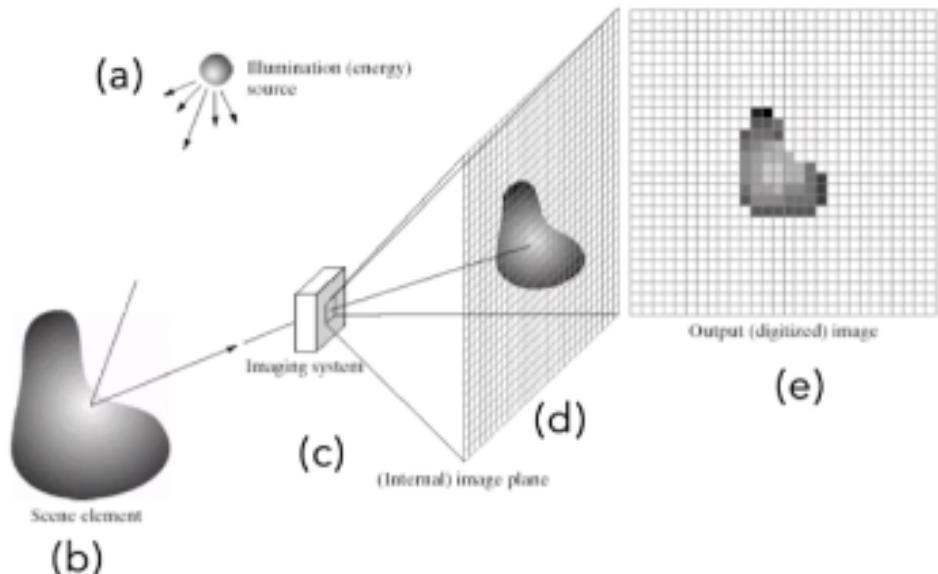
Abbiamo altri dispositivi di imaging, come nel caso delle *tac*: abbiamo un anello con la sorgente luminosa e dall'altro un anello di sensori sensibili ai raggi X e l'oggetto che deve essere acquisito. I raggi X sono assorbiti dai tessuti e non vengono riflessi.

!! Indipendentemente dalla sorgente luminosità, il processo è comune a tutte le sorgenti luminose. !!

---

## DALLA SCENA ALL'IMMAGINE

Abbiamo la sorgente luminosa che illumina la scena, la scena riflette la luce che raggiunge il dispositivo, entra nell'ottica del dispositivo e colpisce il piano immagine del dispositivo che è un array di CCD. Ciò produce un segnale continuo che poi viene presa dal frame grabber e produce la discretizzazione.



## MODELLO DI FORMAZIONE DELL'IMMAGINE

Abbiamo visto come caratterizzare la luminosità dei pixel dell'immagine che dipende da diversi fattori.

Una volta che abbiamo discretizzato l'immagine proveniente dal dispositivo di imaging, quello che otteniamo è un'immagine discreta che è una funzione 2D che a ogni coppia di coordinate che rappresenta le coordinate del pixel, associa il rispettivo valore di intensità che è un valore finito. Il valore di intensità dipende dal modo in cui la scena riflette la luce e l'intensità di ciascun pixel è il prodotto da due componenti:

una componente di illuminazione che ha a che fare con la luce incidente e con una componente di riflettanza. La capacità di riflessione è compresa tra 0 (assorbe tutto) e 1 (riflette tutto).

In pratica:

- $L_{\min} < f(x,y) < L_{\max}$ 
    - Dove:  $L_{\min} = i_{\min} r_{\min}$  e  $L_{\max} = i_{\max} r_{\max}$
  - $[L_{\min}, L_{\max}]$  è chiamato *scala di grigio dell'immagine*
  - Convenzionalmente,  $[L_{\min}, L_{\max}]$  è scalato in  $[0, L-1]$  ([nero, bianco])
    - Sebbene,  $[0,1]$  può essere preferito per scopi di calcolo
- 

## CAMPIONAMENTO E QUANTIZZAZIONE

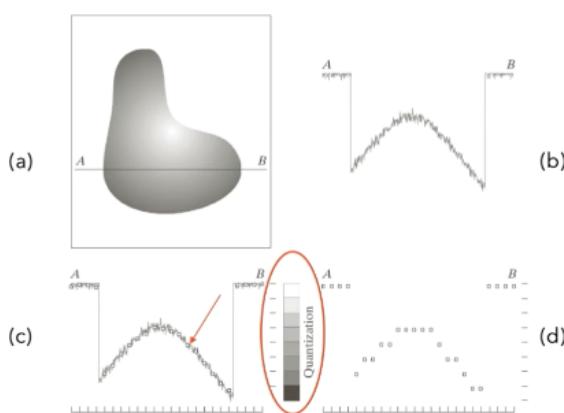
La discretizzazione avviene mediante due fasi:

- Campionamento
- Quantizzazione

Qualsiasi dispositivo di imaging è rumoroso, cioè impreciso.

Per campionare il segnale analogico, lo *spaziamo* sull'asse delle  $x$ . Quanto spaziamo determina quanti campioni andiamo a considerare.

Così abbiamo discretizzato solo con l'asse  $x$ , con la quantizzazione stabiliamo, per una serie di intervalli della variabile di intensità, un valore specifico. Dopo la discretizzazione abbiamo il segnale completamente discreto rispetto ai due assi  $x, y$ .



Maggiore è il numero di pixel dell'immagine e maggiore dovrebbe essere la qualità, ma ciò non è detto.

---

## RAPPRESENTAZIONE DELLE IMMAGINI

Una volta che abbiamo l'immagine discretizzata come la trattiamo?

La trattiamo in 2 modi:

1. Siccome l'immagine è una funzione di 2 variabili, posso rappresentarla in uno spazio cartesiano dove ad ogni coppia  $x$   $y$  associo il valore di intensità. Questo tipo di rappresentazione non è buona.

2. Quello che si va a fare è rappresentarla in un piano 2D. la nostra immagine è una matrice  $m \times n$  dove ciascuna entrata della matrice corrisponde il valore di intensità  $f(x,y)$ .  $f(x,y)$  dopo il processo di discretizzazione può assumere  $L$  valori discreti diversi. Per ragioni

pratiche  $L$  è una potenza di 2 dove  $L = 2^k$ , dove ogni pixel è rappresentato usando  $k$  bit.

L'origine è fissato in alto a sinistra, come nei vecchi schermi della televisione, è comodo poiché le matrici nei linguaggi di programmazione hanno coordinate 0,0 a sinistra.

---

## LIMITAZIONE DEL CAMPIONAMENTO

I dispositivi di imaging sono rumorosi e possiamo osservare degli errori quando rappresentiamo graficamente l'immagine. Questo dipende dalla capacità del dispositivo di rappresentare i diversi livelli di intensità, questa capacità prende il nome di *range dinamico*, l'insieme dei diversi livelli di grigio misurati dal dispositivo di imaging. Possiamo vederlo come il rapporto tra l'intensità massima rilevabile con l'intensità minima rilevabile.

Il valore massimo viene chiamato **saturazione** dove arrivati a un certo livello, tutto viene rappresentato come *bianco*.

Il **contrasto** è definito come la differenza di intensità tra il più alto e più basso livello di intensità di un'immagine. Se ho un'immagine a basso contrasto sarà troppo chiara o troppo scura, se il contrasto è buono avremo un'immagine accettabile.

Il **rumore** è l'errore di misurazione del sensore e può essere individuato nelle regioni più scure dove i pixel invece di essere neri sono grigio scuro.

---

## NUMERO DI BIT PER IMMAGINE

Il numero di livelli di intensità di un'immagine dipende dalla quantizzazione: da quanti valori discreti voglio attribuire all'intensità. Questo valore è una potenza di 2 →  $2^k$ . Se uso un'immagine con un livello di intensità di 256 significa che uso 8 bit.

$N k$	1 ( $L = 2$ )	2 ( $L = 4$ )	3 ( $L = 8$ )	4 ( $L = 16$ )	5 ( $L = 32$ )	6 ( $L = 64$ )	7 ( $L = 128$ )	8 ( $L = 256$ )
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

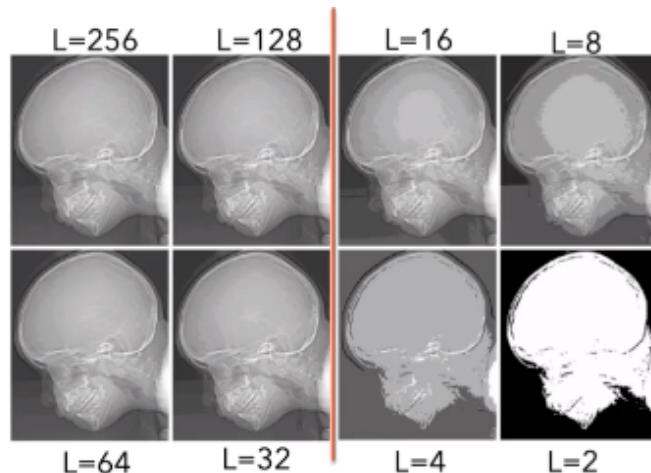
## RISOLUZIONE SPAZIALE

Siamo abituati a dire che «*un'immagine è 2048x2048*», in realtà dire solo questo non ci dice nulla sulla reale risoluzione dell'immagine. Il numero di pixel deve essere associato a un'unità di misura spaziale che dà un'idea dell'immagine, questo concetto deriva dall'editoria. Spesso la risoluzione è misurata in *dpi o ppi*.

## RISOLUZIONE DELL'INTENSITÀ'

La qualità di un'immagine dipende anche dall' intensità: dal numero di livelli di grigio per rappresentare ciascun pixel.

Un basso numero di livelli di grigio in una regione uniforme può causare l'effetto del *falso contorno*.



## INTERPOLAZIONE DI IMMAGINI

Molto spesso è necessario sottoporre le nostre immagini a operazioni di **interpolazione** (data una serie di punti noti, sfruttiamo quelli per determinare punti che non conosciamo) perché le immagini sono sottoposte a **trasformazioni** come rotazioni e le coordinate dei pixel cambiano oppure quando scaliamo un'immagine.

Ci sono 3 tipi di interpolazione:

- 1.Nearest neighbor: prendiamo il valore di intensità più vicino dell'immagine originale.
- 2.Bilineare: prendiamo una combinazione dei 4 più vicini
- 3.Cubica: prendo i 16 più vicini.

Le ultime 2 sono più pesanti computazionalmente ma più precise.

## FUNZIONE DI MAPPING

Supponiamo di avere un'immagine, la cui dimensione è  $M_1 \times N_1$  e la vogliamo riscalare e le dimensioni saranno  $M_2 \times N_2$ . Il rapporto tra queste due immagini quale sarà? Le coordinate []

---

## INTERPOLAZIONE BILINEARE

Vogliamo determinare nella nuova immagine riscalata il valore del pixel nella posizione U.

Prima interpolo rispetto all'asse x e mi trovo il punto R1 e R2, poi interpolo l'asse y e ottengo il valore di P.

Nel caso del Nearest non faccio nessuna interpolazione.

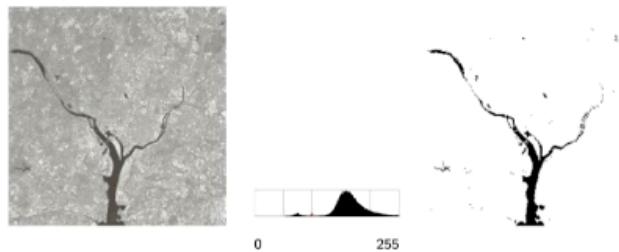
---

## RELAZIONI TRA PIXEL

La nostra immagine è costituita da  $n$  pixel. Come definisco la relazione di vicinanza tra pixel? È importante farlo perché a partire da essa possiamo definire altre relazioni che portano a regioni di interesse di un'immagine per diverse applicazioni.

Innanzitutto dovrei capire quali sono i pixel connessi in un contesto dell'immagine (esempio del fiume).

## ESEMPIO: STIMA DELL'AREA COPERTA DAL FIUME



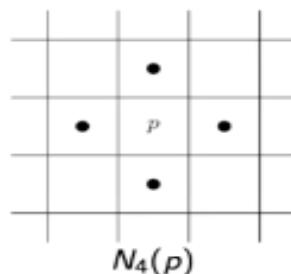
- Identificare i pixel che hanno il colore del fiume
  - Numerose tonalità per il fiume
- Selezionare tra queste i pixel connessi
  - Altre regioni di "acqua" non sono di interesse
- Considerare le regioni che sono "naturalmente" allineate
  - Sezioni dei fiumi suddivise da ponti, restringimenti o coperte

HIST

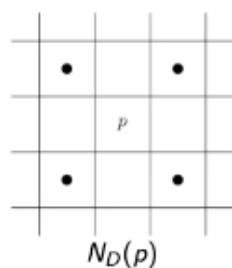
Possiamo definire delle relazioni sui pixel:

### VICINANZA

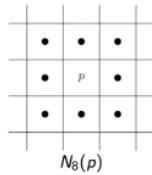
Ho un pixel  $p$  nella posizione  $x$  y, una prima relazione di vicinanza è detta 4-vicinanza. Fissato un pixel chi sono quelli vicini? Sono quelli nelle posizioni



Posso anche definire un'altra vicinanza chiamata *diagonale* dove i pixel vicini sono quelli che si trovano diagonalmente rispetto al punto.



Posso definire anche 8-vicinanza che unisce la diagonale e la 4-vicinanza.



## 2. ADIACENZA

La relazione di adiacenza tiene conto della vicinanza spaziale e del valore di intensità che chiamiamo V.

Definisco 3 misure di adiacenza:

Consideriamo due pixel  $p, q$  che hanno un valore di intensità in un insieme V, i pixel  $p, q$  che hanno valore di intensità in V sono:

**4-adiacenti** se hanno valori di intensità in V e  $q$  è un vicino N4.

**8-adiacenti** se hanno valori di intensità in V e  $q$  è un vicino N8.

**M-adiacenti (mista)** se  $q$  è un vicino N4 oppure è un vicino diagonale tale che l'intersezione tra i vicini N4 di  $p$  e  $q$  deve essere vuota: non devono avere valori in comune. Questa adiacenza può essere utile perché la 8-adiacenza può portare delle ambiguità.

## 3. PERCORSO

Un percorso da un pixel  $p$  al pixel  $q$  è una sequenza di pixel distinti dove il pixel iniziale coincide con  $p$  e quello fine con  $q$ . Se l'inizio coincide con la fine il percorso è chiuso.

Posso avere diversi percorsi a seconda dell'adiacenza.

Il percorso è utile per definire le **COMPONENTI CONNESSE**.

Sia  $S$  un sottoinsieme di pixel di un'immagine e 2 pixel  $p, q$ .

Questi due pixel di  $S$  sono connessi se esiste un percorso da  $p$  a  $q$  che è composto da soli pixel in  $S$ . fissato un pixel  $p$  di  $S$ , l'insieme di pixel connessi a  $p$  prende il nome di **componenti connesse**.

In un'immagine, un qualsiasi sottoinsieme di pixel che è un sottoinsieme connesso è chiamato **regione**.

Dopo aver individuato le regioni, la relazione di adiacenza può essere estesa a esse.

Due regioni sono **adiacenti** se la loro unione è un insieme connesso. Le regioni non adiacenti sono dette **disgiunte**.

L'unione di tutte le regioni di un'immagine è chiamata *foreground*. Il complemento del *foreground* è il *background*.

## BORDO DI UNA REGIONE

Il **bordo** o **contorno** di una regione è l'insieme di pixel della regione che sono adiacenti al background. Il bordo è composto di pixel che hanno almeno un vicino nel background. Tutto dipende dall'adiacenza che adottiamo.

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

Il bordo esterno è sempre un percorso chiuso.

Questa distinzione tra bordo interno ed è esterno è importante perché tutti gli algoritmi di contornazione si basano su un percorso di bordo esterno delle regioni.

L'*edge* (*lato*) è una porzione del contorno, non confondere.

## Lez 7 → FONDAMENTI II: OPERAZIONI SU IMMAGINI

Dobbiamo capire come sono definite le operazioni.

### DISTANZA

Abbiamo definito alcuni concetti usati per diversi algoritmi per individuazione di contorni e lati di un'immagine e abbiamo parlato di **vicinanza**, **adiacenza** e che a seconda del tipo di vicinanza abbiamo diverse nozioni di adiacenza. Ricordiamo che le **regioni** costituiscono gli oggetti di interessi di un'immagine.

Dati 2 pixel  $p$  e  $q$  dobbiamo definire una distanza tra i due pixel. Una **distanza** è una funzione metrica definita su coppie di pixel che gode di 3 proprietà:

1. Funzione non negativa e ritorna 0 se i 2 pixel coincidono

2. Commutatività, la distanza tra  $p$  e  $q$  deve essere uguale alla distanza tra  $q$  e  $p$

$$3. D(p, q) \leq D(p, z) + D(z, q)$$

La distanza euclidea è detta norma  $L_2$ . La distanza euclidea tra 2 pixel è:

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

Abbiamo altri tipi di distanza come **la city block**:

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

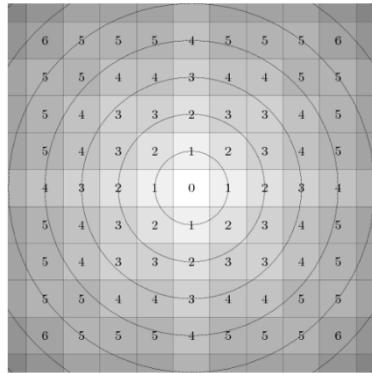
Abbiamo un'altra distanza,  **$D_8$**  che è il massimo tra i valori assoluti delle differenze delle coordinate dei pixel  $p$  e  $q$ .

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

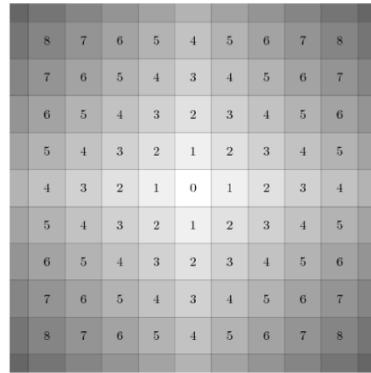
Se fissiamo un pixel  $p$  di coordinate  $x, y$  se vogliamo tutti i pixel a una distanza  $R$ , significa andare a identificare tutti i pixel che si trovano in un cerchio di raggio  $R$ .

Nel caso della distanza city block, se vogliamo tutti i pixel minori di una distanza  $R$ , stiamo identificando un rombo.

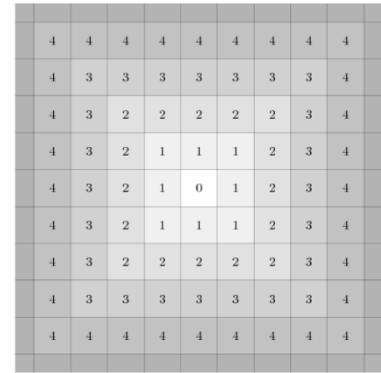
Nel caso di distanza  $D_8$ , stiamo identificando un quadrato.



$D_e$



$D_4$



$D_8$

Quando definisco la distanza, la sto definendo in termini di coordinate spaziali. Non sto definendo una distanza che tenga conto del valore dei pixel, è **importante** fare questa operazione.

Si va a definire la **m-distanza** che è basata sul concetto di m-adiacenza. La m-distanza tiene conto anche del valore dei pixel, devo prima definire un insieme di valore di intensità e poi posso applicare la definizione di **m-path**.

## OPERAZIONI PUNTUALI E MATRICIALI

Le immagini sono rappresentate da **matrici**. Se ho una matrice, posso fare molte operazioni, ma la maggior parte di operazioni sulle immagini sono operazioni **puntuali** e devo definire pixel per pixel.

Ho bisogno di definire il prodotto puntuale di 2 immagini: date 2 immagini, prendo i corrispondenti pixel e ne faccio il prodotto.

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

## OPERAZIONI LINEARI E NON LINEARI

Dobbiamo classificare le operazioni sulla base del tipo di operatori che applichiamo. Gli operatori possono essere lineari o non lineari, la maggior parte

delle teoria dell'elim si basa su operatori lineari, anche se esistono operatori non lineari più efficienti a livello di prestazioni.

Un operatore **lineare**: supponendo di avere un operatore  $H$  applicato a un immagine  $f$  che produce un output  $H[f]$ ,

Se applico questo operatore alla somma delle due immagini moltiplicate per una costante, questa operazione equivale a fare la somma dell'operatore applicato alle singole immagini.

Se un operatore  $H$  soddisfa la proprietà è lineare, oppure non lo è.

La somma è un operatore lineare.

- Operazione lineare
  - Fissato un operatore,  $H$ , che produce un'immagine di output  $g(x,y)$  a partire da un'immagine di input  $f(x,y)$ :  $H[f(x,y)] = g(x,y)$
  - Date due costanti arbitrarie  $a$  e  $b$ , e due immagini qualsiasi  $f_1(x,y)$  e  $f_2(x,y)$ , allora

$$H[af_1(x,y) + bf_2(x,y)] = aH[f_1(x,y)] + bH[f_2(x,y)] = ag_1(x,y) + bg_2(x,y) \quad (1)$$

## ESEMPIO DI OPERATORE NON LINEARE

L'operatore massimo, individua il pixel di intensità massima. Per dimostrare che il massimo è non lineare, consideriamo 2 immagini  $2x2$  con i relativi valori di intensità e consideriamo 2 costanti, applicando la definizione le equazioni non sono soddisfatte.

- 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
- $\max \left\{ \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \right\} + (-1) \max \left\{ \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix} \right\} = 3 + (-1)7 = -4$
  - Quindi, l'operatore di massimo è non lineare

## OPERAZIONI TRA PIXEL

Alcune operazioni tra pixel corrispondono a operazioni matriciali: se considero somma e differenza, ho che l'operazione puntuale corrisponde alla matriciale.

Date due immagini di dimensioni uguali  $MxN$  dove  $x$  varia tra 0 e  $M-1$  e  $y$   $N-1$ :

a-Se voglio fare la **somma** di due immagini devo sommare i corrispondenti pixel.

b.Lo stesso vale per la **differenza**

c.Il **prodotto** e la **divisione** sono stati trattati in precedenza (sopra).

## ES: ADDIZIONE DI IMMAGINI PER RIDURRE IL RUMORE

Ci sono molte applicazioni dove si applicano operazioni come la somma. Un esempio è fatto in astronomia perché spesso si acquisiscono immagini a bassa luminosità che per questo e altri fattori disturbanti, producono immagini rumorose.

In astronomia: se ho delle immagini rumorose e ne faccio la somma, vado a eliminare il rumore.

Bisogna fare delle ipotesi sul rumore: supponendo che il rumore associato alle immagini è **bianco**, si può dimostrare che se ho immagini rumorose, le somme e ne faccio la media, il valore medio della somma corrisponde all'immagine originale senza rumore.

## ESEMPIO: SOTTRAZIONE PER CONFRONTO DI IMMAGINI

È utile quando dobbiamo fare confronti tra immagini per capire come sono variate o quali sono i pixel che sono cambiati rispetto all'immagine originale.

La sottrazione possiamo usarla anche per capire come variando la risoluzione, varia l'intensità di pixel e quali sono i pixel affetti dalla riduzione della risoluzione.

---

## OPERAZIONI INSIEMISTICHE

Per queste operazioni si presume che le immagini siano **binarie**: i valori di intensità sono o bianco o nero.

Per le immagini binarie posso definire insiemi che sono definiti come le coordinate dei pixel in un certo valore.

Se considero un insieme di pixel in un'immagine posso definire le operazioni insiemistiche note.

Ad esempio, abbiamo un'immagine binaria con una serie di oggetti rappresentabili con degli insiemi. Volendo sapere se gli oggetti si sovrappongono basta fare l'intersezione degli insiemi.

Si possono definire anche operazioni su immagini a scala di grigio.

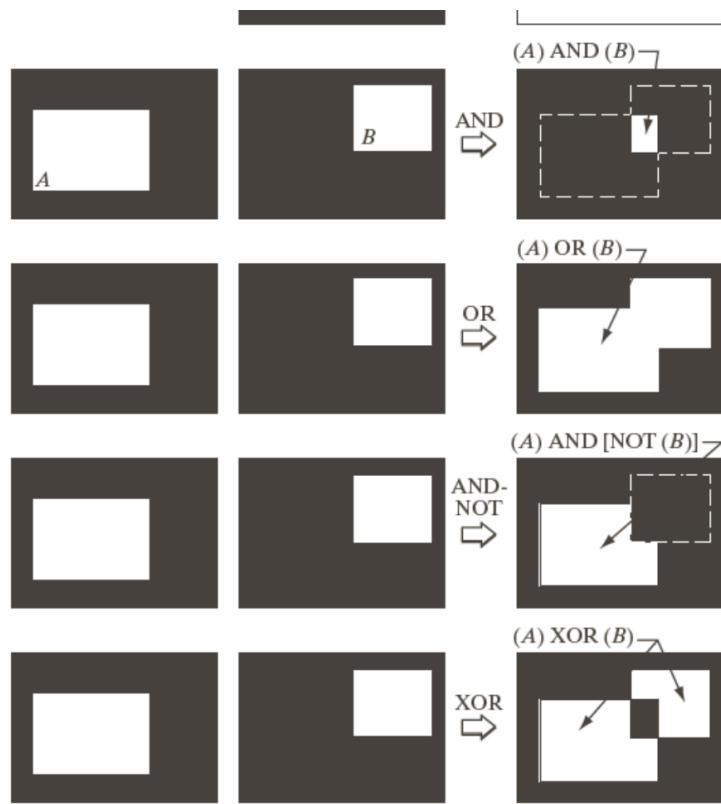
---

## OPERAZIONI LOGICHE

È possibile definire operazioni **logiche**, si presume di avere immagini di nuovo **binarie**. Se interpretiamo i valori dei pixel come **boolean** dove 1 è bianco e 0 è nero, possiamo considerare che tutti i pixel bianchi sono **foreground** e gli scuri

sono **background**. Nel caso di immagini, possiamo fare operazioni logiche tra esse, applicando la definizione.

Ad esempio fare il negato di un'immagine il back diventa foreground...



## OPERAZIONI SPAZIALI

Le operazioni **spaziali** sono eseguite direttamente sui pixel di un'immagine e possiamo classificarle in 3 categorie:

1. Operazioni su singoli pixel: sono operazioni puntuali.
2. Operazioni di vicinanza: ci sono operazioni che mirano a modificare un'immagine ma l'output tiene conto anche dell'intorno di quel pixel.
3. Trasformazioni spaziali geometriche: prendo l'immagine e la modifichiamo geometricamente (ruoto, traslo...).

## OPERAZIONI SUI SINGOLI PIXEL

Presuppongono che ci sia un'immagine da modificare. L'obiettivo è modificare l'aspetto dell'immagine.

Se ho un pixel  $z$ , lo devo trasformare con una funzione.

## OPERAZIONI DI VICINANZA

Invece di prendere due immagini e prendere ciascun pixel e modificarlo, supponiamo di avere parte dell'immagine di input, l'obiettivo è determinare il valore del pixel centrale.

Non vado a modificare il pixel centrale con una funzione ma lo ottengo come la somma dei prodotti del pixel centrale e di un intorno di tutti i suoi pixel vicini. Per fare ciò uso una matrice (**maschera**) della dimensione dell'intorno scelto. Determino il valore del pixel centrale sovrapponendo la maschera e faccio la somma dei prodotti di tutti i pixel dell'intorno per il corrispondente coefficiente

Le operazioni sono abbastanza costose! Se abbiamo una maschera  $3 \times 3$  su un'immagine di  $512 \times 512$ , si necessitano di 5 milioni di operazioni da fare tra moltiplicazioni e addizioni.

Queste sono operazioni locali perché modifichiamo il pixel tenendo conto di una sola regione, chiaramente abbiamo una perdita di informazione: dopo aver modificato l'immagine non possiamo recuperare l'originale.

---

## TRASFORMAZIONI GEOMETRICHE

Sono usate per modificare la dimensione spaziale dei pixel di un'immagine e consistono in due operazioni di base:

1. Trasformazione spaziale delle coordinate

2. Interpolazione che assegna i valori di intensità ai pixel trasformati spazialmente.

Come esprimiamo una trasformazione geometrica? Con un operatore che coinvolge le operazioni matriciali.

- La trasformazione delle coordinate può essere espressa come

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

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

- Esempio:

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

Le operazioni che facciamo sono una serie di trasformazioni dette **affine** (quando le applichiamo preservano punti, rette e piani).

Tutte possono essere rappresentate in matrice tranne la traslazione perché spostiamo le coordinate dei pixel e quindi dobbiamo aggiungere un vettore.

## COORDINATE OMOGENEE

Possiamo rappresentare dei punti all'infinito, come nel caso della proiezione prospettica e quando applichiamo le trasformazioni possiamo avere un'unica matrice che rappresenta una sequenza di trasformazioni (ad esempio se voglio fare *scaling* e rotazione prendo le due matrici e ne faccio il prodotto, ottenendo una sola matrice).

Introducendole, se stiamo nel piano i punti  $x,y$  sono una terna, se siamo nello spazio  $x,y,z$  sono una quaderna.

La matrice è sempre  $3 \times 3$  dove l'ultima riga è sempre  $0 \ 0 \ 1$ .

---

## INTERPOLAZIONE

Una volta che ho modificato le coordinate, siccome non c'è più mapping tra il pixel di input e il corrispondente output, la trasformazione geometrica è seguita dall'interpolazione.

---

## DIMENSIONI IMMAGINI TRASFORMATE

Quando trasformiamo le immagini geometricamente, la rotazione positiva è in senso antiorario. Quando rotiamo un'immagine, l'oggetto in foreground potrebbe uscire dai contorni: o teniamo così l'immagine o la ingrandiamo e mettiamo dei pixel neri di contorno per aggiustarla.

## Lez 8 → TRASFORMAZIONE DI INTENSITÀ'

Oggi vediamo le prime operazioni sulle immagini.

Abbiamo già introdotto le trasformazioni di intensità, dicendo che le immagini sono una matrice dove ogni entrata indica l'intensità del pixel corrispondente. Possiamo manipolare le immagini, operando delle trasformazioni che possono essere di due tipi, soprattutto se parliamo nel dominio spaziale:

**1.Puntuale:** modifica pixel per pixel.

**2.Filtraggio:** modifico i valore dei pixel ma invece che ad ogni pixel di input ci sia un output trasformato, modifico un pixel più il suo intorno che prende il nome di *kernel o maschera*. Tipicamente è un quadrato di dimensioni dispari.

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

L'obiettivo delle trasformazioni puntuali è modificare le immagini perché un dispositivo di acquisizione ha prodotto immagini poco qualitative per produzione di rumore, troppa o poca luce, lenti non efficienti...

In questo modo facilitò l'interpretazione dell'esperto di un dominio (es un medico che vede una radiografia) oppure perché le immagini possono essere date in input a sistemi AI che necessitano immagini di buona qualità.

Le trasformazioni spaziali (filtraggio) sono effettuate perché spesso le immagini trasformate verranno successivamente manipolate con altre operazioni come *sharpening o blurring*.

Un motivo per cui si fa una trasformazione di intensità è che abbiamo un'immagine scura e voglio aumentare il contrasto per avere un aspetto più nitido.

Per introdurre queste tecniche indichiamo le immagini di input con  $f(x,y)$  dove a ogni coordinata  $x,y$  ho dei valori. Quando ho un'immagine di input e voglio trasformarla con una funzione  $T$  ottengo un output  $g(x,y)$ .

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

Se consideriamo il singolo pixel allora parliamo di trasformazione di intensità.

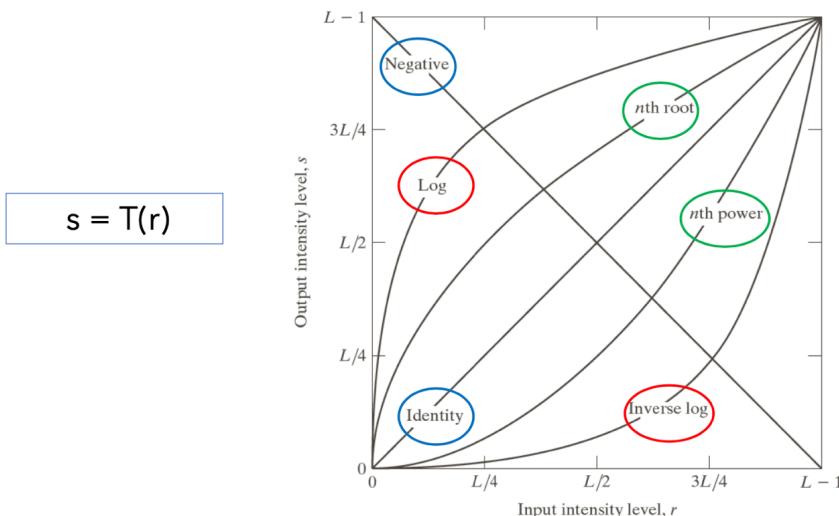
Abbiamo un'immagine di input con i suoi valori che chiamiamo  $R$  e ottengo in output un'immagine dove il pixel corrispondente è dato da una trasformazione applicata al corrispondente pixel di input.

Le trasformazioni di intensità sono abbastanza semplici da implementare: si usano delle tabelle che contengono i valori di intensità dei pixel di I/O e quando devo determinare il corrispondente output del pixel di input faccio una ricerca nella tabella.

Posso applicare una serie di trasformazioni a immagini di qualsiasi banda: raggi  $x, \gamma \dots$  una trasformazione che applico in una banda visibile potrebbe non dare risultati desiderati su un'altra banda.

In generale non c'è una regola di ottimalità: a seconda dei bisogni devo usare una trasformazione adatta anche se non è certo che questa migliori l'immagine.

## TRASFORMAZIONI DI INTENSITÀ



iT

Le trasformazioni *log* sono logaritmiche, consentono di estendere il contrasto dove pixel scuri diventano più chiari.

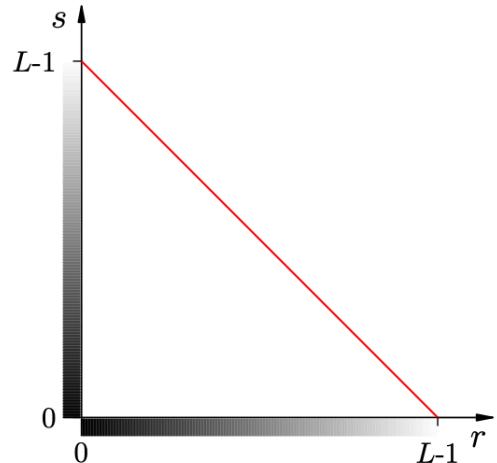
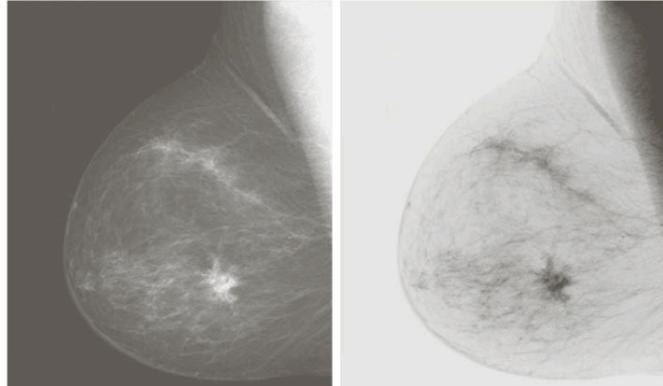
La logaritmica inversa fa l'inverso.

Le *negative* trasformano il pixel da bianco a nero e vice.

### NEGATIVA DI UN'IMMAGINE

Per ogni pixel di input il cui valore di intensità è  $r$  corrisponde in output il pixel  $s$  la cui intensità è  $(L-1) - r$  dove  $L$  è il numero dei livelli di grigio.

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

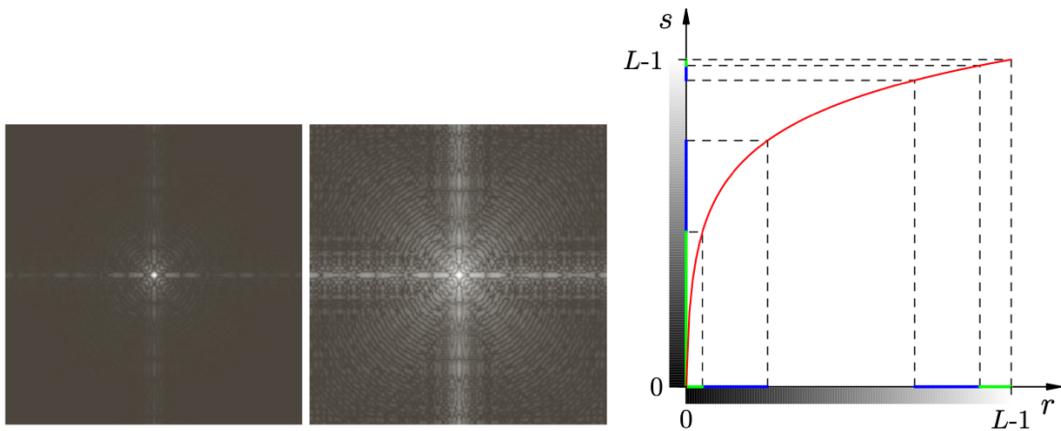


## TRASFORMAZIONI LOGARITMICHE

Il valore di ogni pixel di output è ottenuto dal pixel di input applicando la seguente funzione dove  $c$  è una costante:

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

Per pixel quasi neri, faccio corrispondere una gamma di pixel più chiari. Se ho un'immagine scura ma ho pochi pixel estremamente chiari, se applico la trasformazione logaritmica do la possibilità ai pixel più scuri di rendersi più visibili. L'esempio mostra la rappresentazione dello spettro di *Fourier*: *dopo averla trasformata ottengo un'immagine molto più chiara*. I valori dei pixel che erano chiari ma erano invisibili quasi, ora sono più nitidi.



## TRASFORMAZIONI $\gamma$

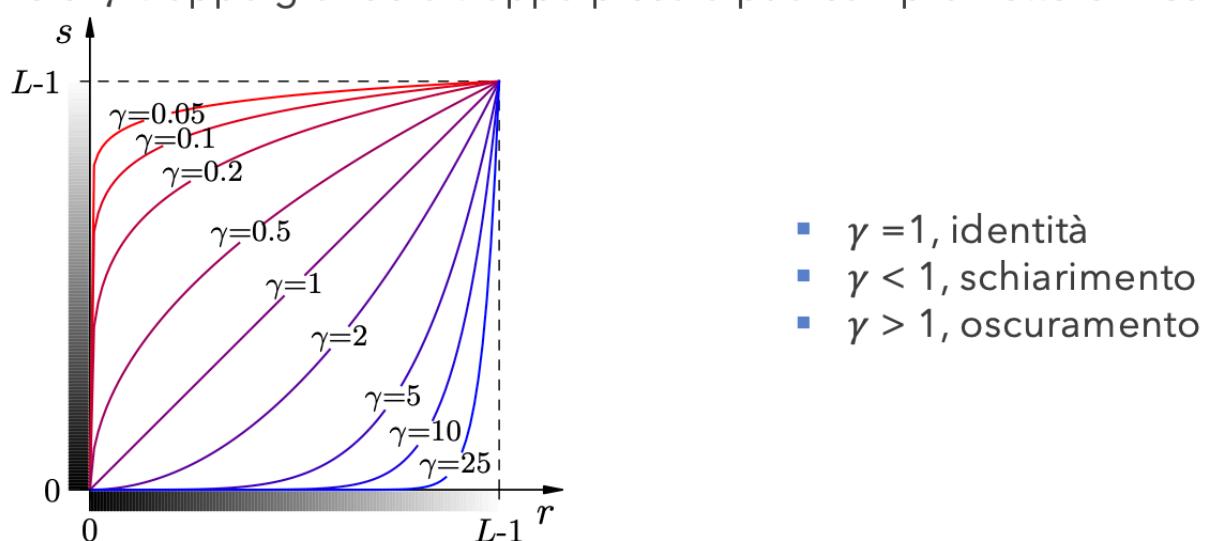
Chiamate anche trasformazioni di **potenza**.

- $\mathbf{s} = \mathbf{c}\mathbf{r}^\gamma$  (talvolta  $\mathbf{s} = \mathbf{c}(\mathbf{r} + \boldsymbol{\varepsilon})^\gamma$ , dove  $\mathbf{c}$  e  $\gamma > 0$

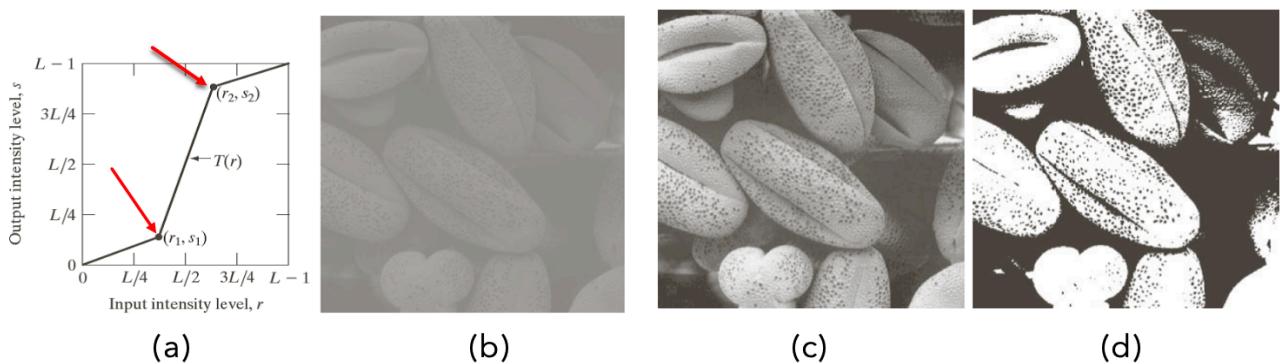
$\gamma$  è un coefficiente arbitrario da impostare per il risultato che vogliamo, lo stesso  $\mathbf{c}$ .

Si chiama  $\gamma$  poiché deriva dalla procedura di correzione che si applica ai dispositivi di visualizzazione: i monitor *crt* avevano una risposta che era alterata rispetto al segnale ottenuto poiché le immagini erano più scure: per migliorarle si applica la trasformazione  $\gamma$ . Oltre a migliorare le risposte dei dispositivi, posso usarle per migliorare l'immagine.

Il comportamento dipende quindi dal valore assegnato a  $\gamma$ : se gamma è minore di 1, l'effetto è simile alla trasformazione logaritmica (migliora il contrasto dell'immagine). Se  $\gamma$  è maggiore di 1 ha l'opposto (scuriamo l'immagine). Se  $\gamma$  è 1 c'è identità.



## TRASFORMAZIONI DI STRECHING DEL CONTRASTO



Possiamo fare altre trasformazioni dove se rappresentate in termine di funzione possono avere un qualsiasi aspetto. Sono trasformazioni lineari a pezzi.

Devo definire coppie di punti le cui coordinate determinano l'aspetto della trasformazione.

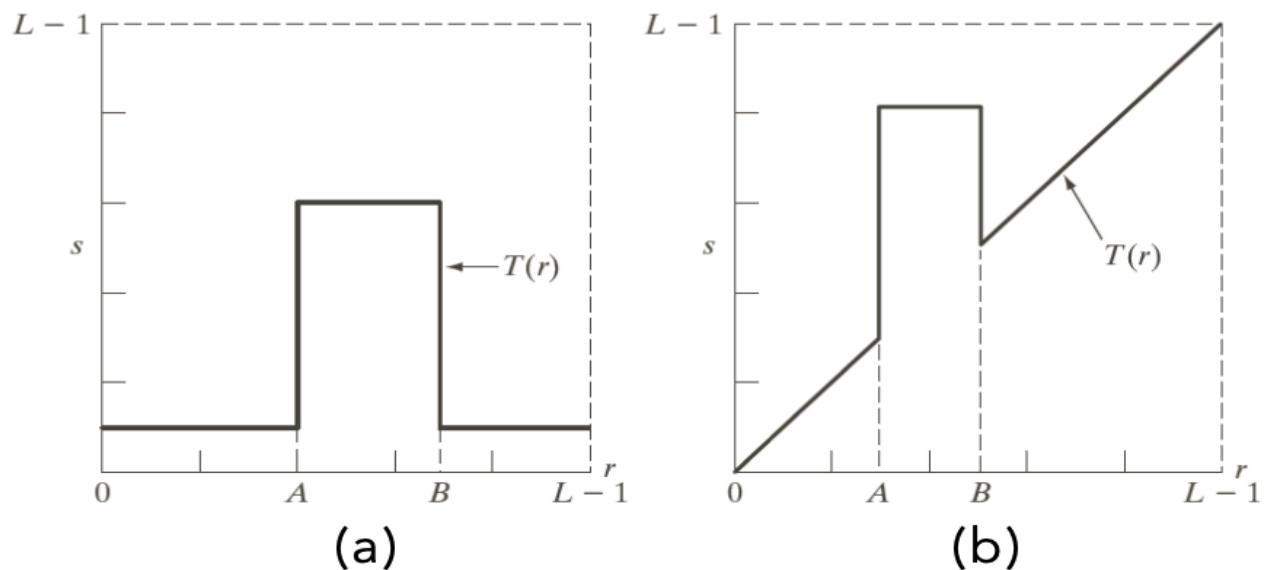
Ad esempio nell'immagine i pixel fino a un certo range sono mappati e oltre a questo punto si ha uno *stretching* per tutti i valori. Se porto lo stretching all'estremo, **binarizzo** l'immagine facendola diventare bianca e nera.

### SLICING DEL LIVELLO DI INTENSITA'

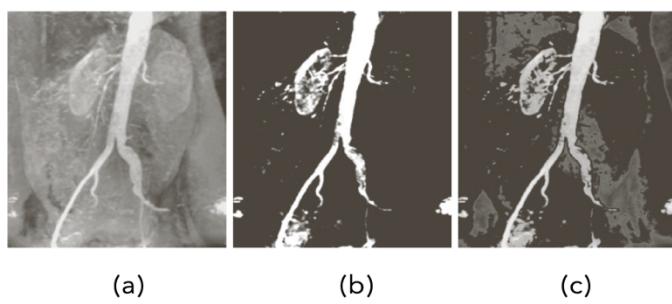
Creo range di intensità da sottoporre alle trasformazioni.

Se ho un range che voglio evidenziare, per quel range mappo rendendo tutto chiaro e al di fuori di esso faccio diventare tutto scuro.

Oppure tutto ciò che non sta nel range non interessato rimane inalterato.



- (a) Immagine originale
- (b) I vasi sono evidenziati impostando a  $L-1$  i livelli di intensità che sono nel range di interesse e a 0 tutti gli altri
- (c) Le intensità dei vasi sono conservative, mentre le altre sono scurite



Nell'esempio vediamo che grazie allo *slicing* posso evidenziare regioni che interessano maggiormente.

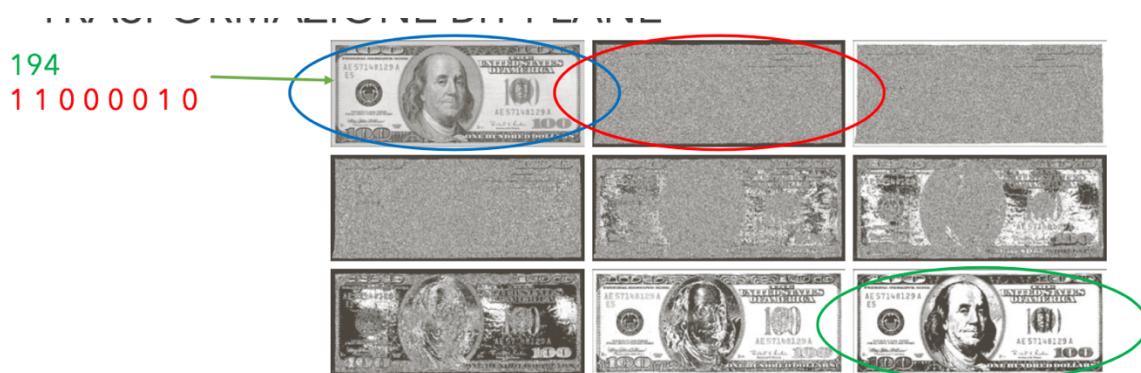
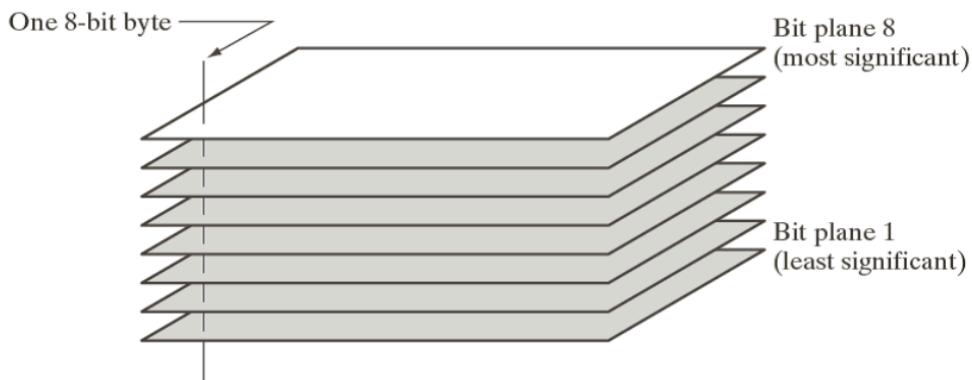
## BIT-PLANE SLICING

Questo slicing non è una trasformazione di intensità vera e propria. Si chiama bit plane slicing e serve a capire se il numero di bit usati per la quantizzazione è adeguato e per la compressione di immagini.

Invece di considerare l'immagine come una matrice di interi, posso vederla come composta da strati di bit.

Posso fare 8 tagli (se ho 8 bit) con il più significativo, il secondo più significativo e così via...

Faccio questo perché posso vedere qual è il contributo di ciascun bit con cui rappresento i valori di intensità.



- Ogni strato contribuisce all'aspetto finale dell'immagine, ma il più dell'informazione è negli strati più alti



In questo modo vedo che i bit più significativi rendono l'immagine chiara. Posso inoltre stabilire quanti bit mi servono per mantenere la stessa info dell'immagine di partenza. Per fare questo posso sovrapporre il *plane* del più significativo con il secondo più significativo e sommarli e così via...

## TRASFORMAZIONE BIT-PLANE

- Immagini ottenute usando
  - (a) i bit-plane 8 e 7
  - (b) i bit-plane 8, 7 e 6
  - (c) i bit-plane 8, 7, 6 e 5
- Oss: si moltiplicano i pixel del piano n-esimo con la costante  $2^{n-1}$



Con abbiamo capito che invece di usare 8 bit potrei usarne solo 4, così facendo ho compresso l'immagine.

### ISTOGRAMMA DI UN'IMMAGINE

Passiamo a trasformazioni basate sul concetto di istogramma di un'immagine.

Per avere un istogramma di un'immagine conto il numero di valori dei pixel (es quanti pixel neri, bianchi ci sono). L'istogramma dà info sulle immagini utili. L'istogramma di un'immagine a L-valori è una funzione discreta:

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

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

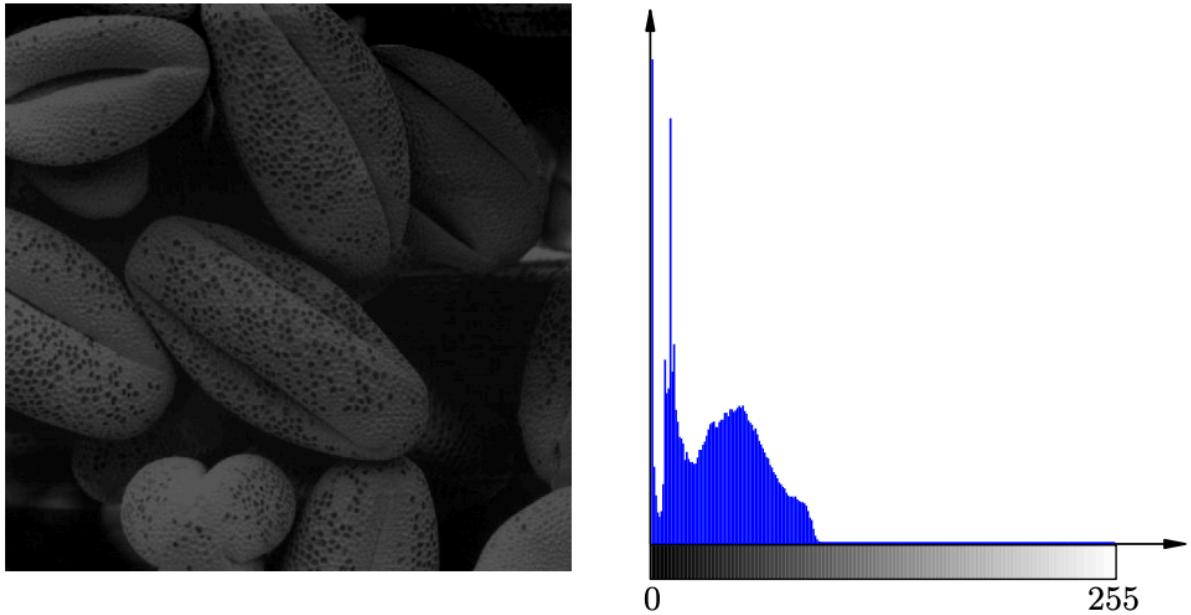
Se normalizzo l'istogramma posso interpretare questi conteggi come delle probabilità.

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

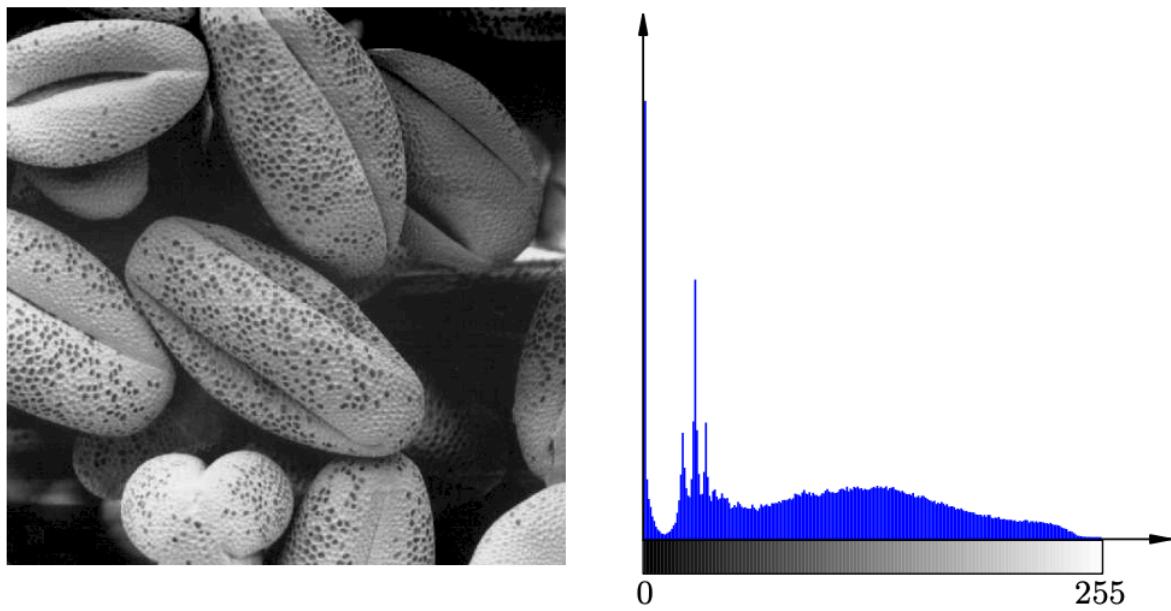
$p(k)$  è la probabilità con cui un pixel assume l'intensità  $k$ .

Se faccio la somma dei  $p(k)$  alla fine ottengo 1.

COSA CI DICE L'ISTOGRAMMA?



Calcolando l'istogramma noto che ho pixel scuri tendenti a 0.  
 Se ho pixel chiari tendono a sinistra, se invece sono a basso contrasto li ho al centro.  
 Per migliorare l'immagine devo fare in modo che i punti siano distribuiti sul range di tutti i valori, questo processo è detto *equalizzazione dell'istogramma*.  
 L'ideale è:



## TRASFORMAZIONI MONOTONICHE

Abbiamo una trasformazione continua, se presi 2 valori della variabili di input:  $r_1$  e  $r_2$ ,  $r_2$  deve essere maggiore di  $r_1$ .

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

$$T(r_2) \geq T(r_1), \quad r_2 > r_1 \quad \text{Mononota non descrescente}$$

$$0 \leq T(r) \leq L - 1, \quad 0 \leq r \leq L - 1$$

Se la trasformazione è monotona strettamente crescente, è importante perché posso invertire la trasformazione e risalire dall'output al valore di partenza.

I valori di intensità possiamo vederli come variabili casuali, se faccio ciò e ho una trasformazione continua e differenziabile.

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

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

- In particolare, è interessante la seguente trasformazione:

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

- Allora:

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

- Quindi:

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

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

In breve, se ho un'immagine e determino che ha una distribuzione  $p(r)$  se applico una trasformazione come quella in figura, ottengo un output con pixel che hanno una distribuzione uniforme.

## EQUALIZZAZIONE DI UNA VAR CASUALE DISCRETA

- $r_k$  è il livello di intensità in  $0, \dots, L-1$ 
  - $p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, \dots, L-1$
- Un'immagine può essere equalizzata assegnando l'intensità  $s_k$  a quei pixel che hanno intensità  $r_k$  tramite:

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

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

Dopo aver definito la probabilità su ogni pixel devo equalizzare, applicando la trasformazione precedente.

Trasformando così i pixel dell'immagine di input otteniamo un output i cui pixel approssimativamente sono equamente distribuiti.

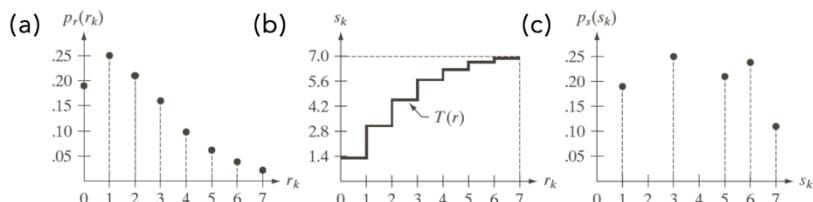
## ESEMPIO

Supponiamo

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

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

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



ELIM a.a. 2020/2021 - Parte I - Prof. A. Staiano

$T(r_k) = s_k$  ma approssimato all'intero più vicino.

$P_s(s_k)$  è dato da  $\frac{n_k}{4096} + \text{gli } n_k \text{ precedenti}$

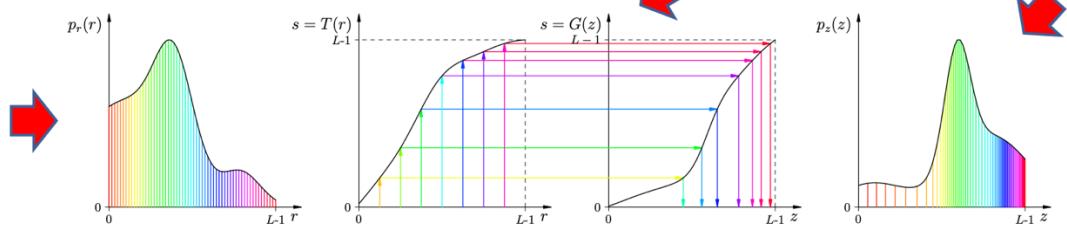
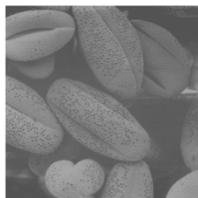
L'equalizzazione non funziona sempre: se ho un'immagine con pixel molto scuri e altri comunque tendenti al nero, i valori scuri diventano chiari ma non ho migliorato il contrasto.

## SPECIFICA DELL'ISTOGRAMMA

Grazie alla manipolazione dell'istogramma posso migliorare l'aspetto di un'immagine. Ci sono applicazioni in cui non voglio una densità uniforme ma una ben precisa perché posso sfruttare conoscenze sull'immagine per migliorarla. Voglio fare una trasformazione che in output produca una trasformazione data in input. Questo è detto *matching dell'istogramma*: oltre all'input do anche la densità di probabilità in output.

## SPECIFICA DELL'ISTOGRAMMA

- Sia  $s$  un variabile casuale tale che:
  - $s = T(r) = (L-1) \int_0^r p_r(w)dw$
  - $p_s$  è uniforme
- Definiamo una funzione  $G$  su  $z$  che soddisfa:
  - $G(z) = (L-1) \int_0^z p_z(t)dt = s$
- Quindi:  $G(z) = s = T(r)$
- Il mapping cercato  $F$ , tale che  $z = F(r)$  può essere ottenuto come:
  - $z = G^{-1}(T(r))$



Dalle equazioni rossa e verde possiamo ottenere un'immagine i cui livelli di intensità abbiano una pdf specificata come:

1. Si usa la rossa per ottenere i valori di  $s$ .
  2. Si usa la blu per ottenere  $G(z)$
  3. Si ricava la verde ottenendo una trasformazione da  $s$  a  $z$
  4. Si ricava l'immagine di output equalizzando l'immagine di input tramite la rossa:
    - i valori dei pixel sono i valori  $s$
    - Per ogni pixel con valore  $s$  dell'immagine equalizzata, si applica la trasformazione inversa verde per ottenere i pixel dell'immagine di output
- 
- Una volta manipolati tutti i pixel la pdf dell'immagine di output sarà approssimativamente uguale a quella trattata

- Le equazioni **rossa** e **verde** possiamo ottenere un'immagine i cui livelli di intensità abbiano una pdf specificata come:
  1. Si usa la **rossa** per ottenere i valori di  $s$
  2. Si usa la **blu** per ottenere  $G(z)$
  3. Si ricava la **verde** ottenendo una trasformazione da  $s$  a  $z$
  4. Si ricava l'immagine di output equalizzando l'immagine di input tramite la **rossa**
    - i valori dei pixel sono i valori  $s$
    - Per ogni pixel con valore  $s$  dell'immagine equalizzata, si applica la trasformazione inversa **verde** per ottenere i pixel dell'immagine di output
- Una volta manipolati tutti i pixel la pdf dell'immagine di output sarà approssimativamente uguale a quella trattata

## SPECIFICA ISTOGRAMMA CASO DISCRETO

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

- Similmente, dato  $s_k$ ,

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

- Infine,

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

- Non calcoliamo  $G^{-1}$  ma calcoliamo tutti i valori di  $G$  dalla blu, per  $q=0, \dots, L-1$ 
  - Memorizzati in una tabella
  - Per ogni  $s_k$  cerchiamo il valore nella tabella,  $z_q$ , più vicino (soluzione migliore approssimata)

## SPECIFICA DELL'ISTOGRAMMA

Ottengo l'istogramma dall'immagine di input.

Faccio l'equalizzazione e ottengo i valori di  $s_k$ .

3. Ottenerne il mapping  $G$  dalla  $p_z$  specificata  $G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i)$  (arrotondando  $z_q$  a numeri in  $[0, \dots, L-1]$ )
4. Per ogni  $s_k$ ,  $k=0, 1, \dots, L-1$ , usare i valori di  $G$  in tabella per trovare i  $z_q$  corrispondenti, in modo che  $G(z_q)$  è più vicino a  $s_k$ 
  1. Se ci sono più valori  $z_q$  per cui  $G(z_q)$  è il più vicino a  $s_k$ , scegliere il più piccolo
5. Applicare la trasformazione  $F$  all'immagine originale tramite in mapping trovato al passo 4

## ELABORAZIONE LOCALE DELL'ISTOGRAMMA

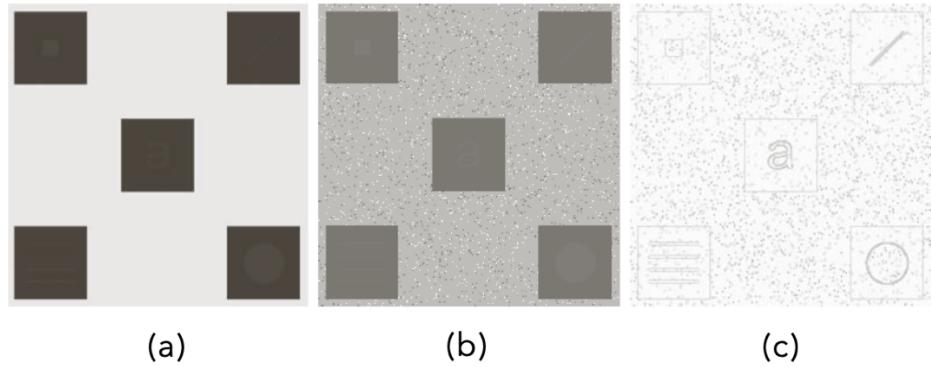
Quando equalizzo un istogramma, ho un approccio globale poiché tengo conto di tutti i pixel dell'immagine.

Può succedere che dei pixel possono avere valori vicini a una regione scura ma il loro numero è basso che durante l'equalizzazione non si mettono in evidenza, anche se potrebbero dare info importanti.

Quando è così, invece di fare l'equalizzazione globalmente, lo facciamo solo per un'area di interessa che tiene conto dell'intorno della zona di interesse.

Se prendo regioni  $3 \times 3$  e applico la quantizzazione posso ottenere risultati migliori.

## ESEMPIO



(a) immagine originale

(b) immagine equalizzata

(c) immagine equalizzata localmente (vicinanza 3x3)

## STATISTICHE DA ISTOGRAMMA

Dagli istogrammi posso calcolare molti indici statistici, su un'intera immagine o in regioni di pixel:

- 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:  $\mu_n = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$

Posso lavorare su questi indici per migliorare l'immagine localmente.

## ESEMPIO

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

## Lez 9 → Trasformata di Fourier

Le operazioni di filtraggio possono essere fatte anche nel dominio delle frequenze: gli output sono gli stessi del dominio spaziale, cambia solo a livello di complessità computazionale e lavorare nel dominio delle frequenze risulta più “naturale”.

Lo strumento fondamentale per lavorare nel dominio delle frequenze è la *trasformata di Fourier* (*evitiamo passaggi e dimostrazioni ma gli aspetti matematici sono importanti*).

[all’orale chiede le formule e non dimostrazioni].

---

### FUNZIONI TRASFORMATE

Le trasformate sono usate perché nei domini le operazioni possono essere più convenienti o semplificate.

Se ho una funzione e devo fare delle operazioni, prima di farlo mi sposto in un altro dominio, faccio le trasformate e poi torno al dominio originale con una funzione inversa.

I domini di partenza sono quello dello spazio (funzioni la cui variabile indipendente rappresenta lo spazio o il tempo) e il dominio trasformato è quello delle *frequenze*.

L’immagine è di fatto un segnale, ciascun segnale può essere rappresentato come una somma di funzioni con una certa frequenza.

---

### JOSEPH FOURIER

Fourier scoprì che una qualsiasi funzione periodica può essere espressa come una somma di funzioni seno e coseno di frequenze diverse, ciascuna moltiplicata con un diverso coefficiente.

Tutto ciò va sotto il nome di *Serie di Fourier*.

Dimostrò anche le funzioni periodiche possono essere espresse come somme di funzioni seno e coseno, che prende nome di *Trasformata di Fourier*.

---

### NUMERI COMPLESSI

Un numero complesso è costituito da 2 termini:

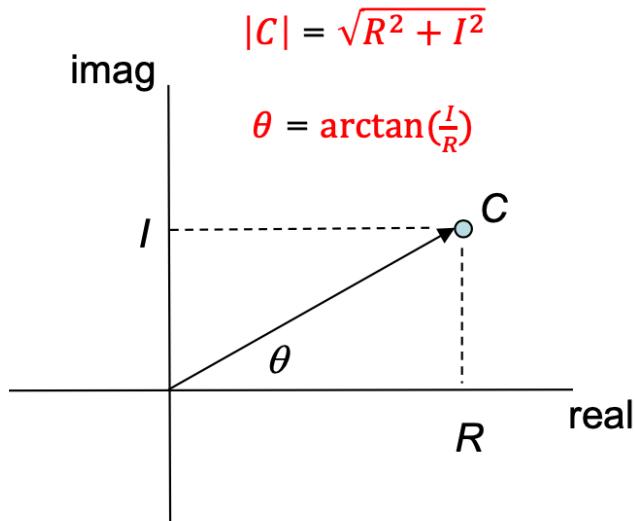
Una parte immaginaria

Una parte reale

- Sia  $j$  il numero immaginario  $j = \sqrt{-1}$
- Un numero complesso è definito come:  $C = R + jl$
- Complesso coniugato:  $C^* = R - jl$
- In coordinate polari:  $C = |C|(\cos \theta + j \sin \theta)$  o  $C = |C| e^{j\theta}$

I numeri complessi si possono rappresentare in un piano come un vettore: la parte reale sulle ascisse e la parte immaginaria sulle ordinate. L'angolo tra le ascisse e il vettore è detto *theta*.

Dato un numero complesso possiamo definirne il modulo e l'angolo.



Possiamo esprimere il numero complesso in coordinate polari.

---

## SERIE DI FOURIER

Ogni funzione periodica,  $f(t)$ , con periodo  $T$ , può essere rappresentata come combinazione lineare di **seni** e **coseni**.

È usata per approssimare le funzioni periodiche.

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

dove

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

- Osservazione: la base è composta di un insieme infinito di seni e coseni
- 

## IMPULSO

Altri elementi utili per le operazioni nel dominio delle frequenze è l'impulso. L'impulso è il valore di un segnale che improvvisamente raggiunge il punto massimo ed è sempre zero dopo.

L'impulso è definito come:

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

L'impulso massimo è a zero ed è zero negli altri punti. Se considero il suo integrale è 1.

Non esiste una funzione realistica che abbia tali proprietà.

L'impulso è definito nel punto 0 ma possiamo definirlo anche in un punto arbitrario  $T_0$ .

L'impulso ha una proprietà importante chiamata *sifting*: se integro il prodotto tra una funzione e l'impulso, il risultato è il valore della funzione nel punto dell'impulso:

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


---


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

### TRENO D'IMPULSO

Posso definire un treno di impulsi, cioè una sequenza di impulsi che si ripetono con un certo periodo.

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

### TRASFORMATA DI FOURIER CONTINUA

Per ogni funzione continua  $f(t)$ , la trasformata di *Fourier di  $f(t)$*  è definita come:

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

è l'integrale della funzione originale che moltiplica la componente immaginaria  $* 2\pi i \nu t dt$ .

Poiché è integrata in  $t$ , scompare e la funzione dipende solo da  $\nu$ , che è la frequenza.

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

Con questa trasformata di Fourier trasformo la funzione dal dominio di partenza a quello delle frequenze mentre con l'inversa torno al dominio originale:

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

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

[imparare le formule]

---

## FUNZIONE SINC

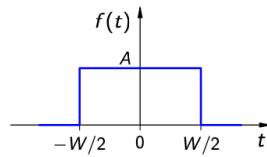
La funzione **sinc** vale 1 in zero e  $\sin \pi t / \pi t$  altrimenti.

La trasformata di Fourier di una funzione **box** è la **sinc**.

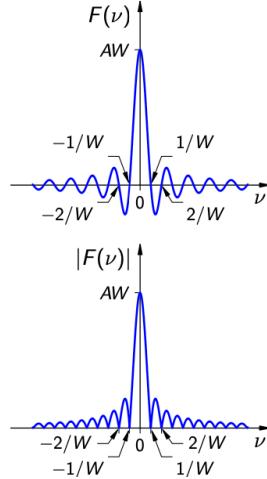
Le funzioni sono inversamente proporzionali.

## ESEMPIO: TF DELLA FUNZIONE BOX

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



$$\begin{aligned} F(\nu) &= \int_{-\infty}^{\infty} f(t) e^{-i2\pi\nu t} dt \\ &= \int_{-W/2}^{W/2} A e^{-i2\pi\nu t} dt \\ &= AW \frac{\sin(\pi\nu W)}{\pi\nu W} \\ &= AW \operatorname{sinc}(\nu W) \end{aligned}$$



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

La trasformata di un impulso è una costante:

$$\mathcal{F}(\delta(t)) = F(\nu) = \int_{-\infty}^{\infty} \delta(t) e^{-\imath 2\pi \nu t} dt = e^{-\imath 2\pi \nu 0} = 1$$


---

## CONVOLUZIONE E TRASFORMATO DI FOURIER

Quando si parla di operazioni di filtraggio si introducono *correlazione e convoluzione*.

Possiamo definire la convoluzione di due funzioni continue,  $f$  e  $h$ , definita come:

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

- Può essere mostrato che:

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

- Anche l'opposto vale:

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

- Questo è chiamato **teorema di convoluzione**

Il teorema di convoluzione lega il dominio spaziale con quello delle frequenze in termini di convoluzioni. Se sono nel dominio originario e voglio fare la convoluzione di due funzioni posso anche fare il prodotto delle trasformate o se voglio fare il prodotto di due funzioni nel dominio originario mi basta fare la convoluzione.

## CAMPIONAMENTO

Nel momento in cui c'è un segnale continuo da discretizzare, perdo informazioni poiché esamino solo dei punti e a seconda di quanti si considerano ho una rappresentazione più o meno fedele.

Quando lavoriamo coi segnali di qualsiasi tipo (immagini, audio), quando discretizziamo perdiamo informazioni, per poterlo ricostruire fedelmente senza artefatti, quanti campioni devo prendere?

La trasformata di Fourier ci aiuta in questo contesto.

Ho una funzione continua  $f(t)$ , per discretizzarla devo prendere i valori ogni  $n$  passi: come faccio? Prendo la funzione e la modifco per un treno di impulsi:

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

*deltaT* è il periodo del treno di impulso che indica gli n campioni.

Se voglio il valore del campione in corrispondenza dell'impulso basta applicare la definizione:

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

## TRASFORMATA DI FOURIER DI UNA FUNZIONE CAMPIONATA

Usando il teorema di convoluzione, la trasformata di Fourier di una funzione campionata è calcolata come:

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

- Dalla proprietà di sifting di  $\delta$  :

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

Questa formula ci dà molte info:

1. Sebbene  $f(\tilde{t})$  è una funzione campionata,  $F(\tilde{\nu})$  è continua.
2. È la somma di copie scalate di una funzione continua  $F$

- La separazione tra le copie è regolata da  $1/\Delta T$ :

Sovra campionamento:  $1/\Delta T$  ha un valore per cui la separazione è evidente

Campionamento critico

Sotto campionata:

## TEOREMA DI CAMPIONAMENTO

Una funzione  $f(t)$  è detta a *banda limitata* se la sua trasformata è 0 per valori di frequenza esterni a un intervallo detto **banda**.

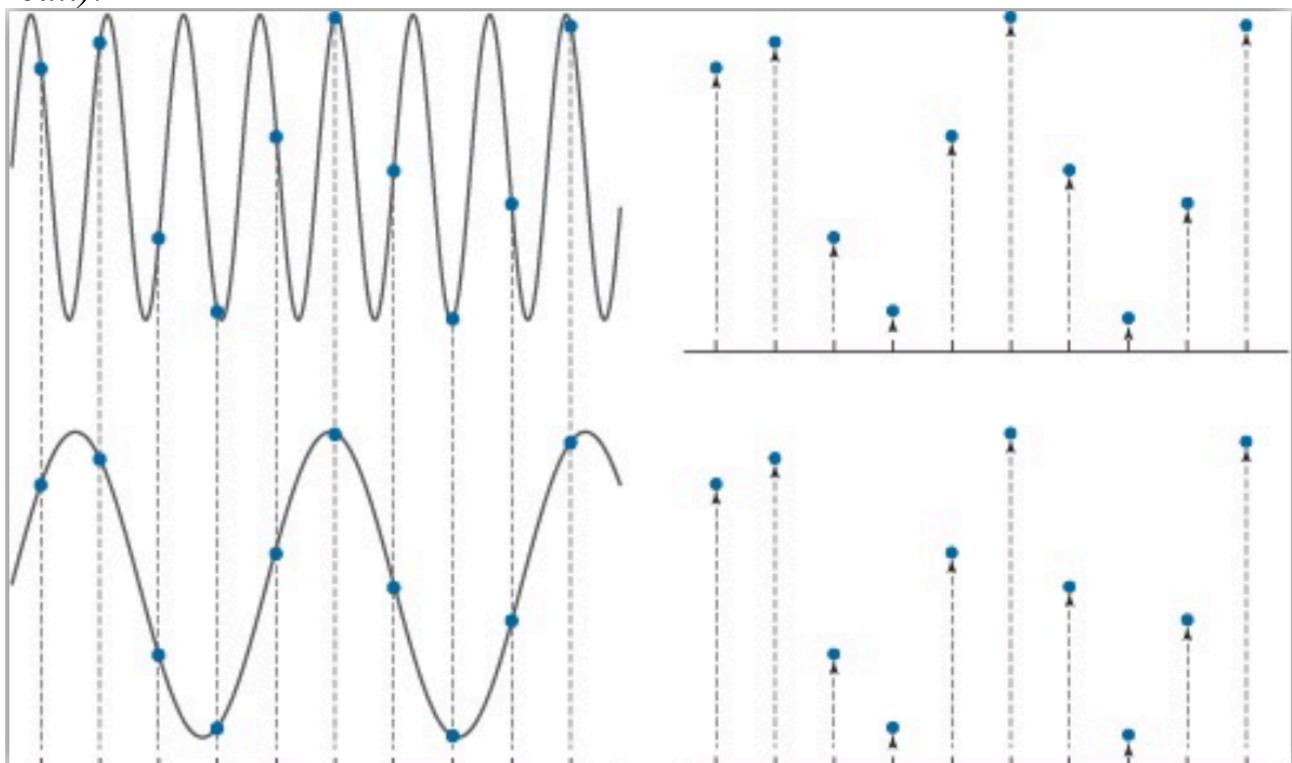
Se ho una funzione a banda limitata, posso ricostruire la funzione originaria andando nel dominio delle frequenze isolando una singola copia. È isolabile la copia solo quando si può distinguere dalle altre.

- La condizioni precedenti possono essere riepilogate nel **teorema di campionamento**:  $1/\Delta T > 2 v_{max}$
  - Il tasso di campionamento deve essere almeno il doppio della frequenza massima in  $f(t)$
  - Il tasso di campionamento critico è chiamato **tasso di Nyquist**
- 

## ALIASING

Significa avere una diversa identità per una stessa cosa.

Se campiono senza rispettare il tasso di nyquist si verifica il segnale di **aliasing** (*nelle immagini compaiono artefatti non esistenti, nell'audio sono suoni non reali*).



Le due funzioni hanno la stessa trasformata e se faccio l'inversa non riesco a risalire all'originale poiché non saprei a chi attribuirla.

L'aliasing è evitabile se si rispetta il tasso di nyquist ma non è sempre così purtroppo.

Se lavoriamo con un segnale audio/video non sono funzioni illimitate e l'aliasing infatti è quasi sempre inevitabile. Gli strumenti di discretizzazione applicano un filtro di smoothing al segnale originale per ridurre le frequenze elevate e ridurre l'aliasing.

---

## RICOSTRUZIONE DI UNA FUNZIONE CAMPIONATA

Moltiplichiamo la funzione per la funzione box, per ottenere un periodo di F.

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

- Ovvero,  $F(\nu) = H(\nu) \tilde{F}(\nu)$
- Allora, può essere applicata la trasformata di Fourier inversa per ottenere f

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

- Può essere mostrato che:

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

- f(t) è uguale a fk in t = k ΔT
- Altrove è ottenuta interpolando le funzioni di base sinc shiftate
  - In pratica, facciamo interpolazioni tra numero finito di campioni e quindi otteniamo un'approssimazione
    - Nell'equazione dovremmo interpolare tra infiniti campioni

---

## TRASFORMATA DI FOURIER DISCRETA (DFT)

Le nostre immagini o audio sono funzioni discrete e non continue quindi dobbiamo ottenere le espressioni corrispondenti per funzioni discrete, anche la stessa trasformata deve avere una controparte discreta.

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

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

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

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

- Da cui può essere mostrato:

$$\tilde{F}(\nu) = \sum_{n=-\infty}^{\infty} f_n e^{-i 2 \pi \nu n \Delta T}$$

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

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

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

- E i campioni sono:

$$F_m = \sum_{n=0}^{M-1} f_n e^{-\imath 2\pi mn/M}$$

- Gli M campioni  $\{F_m\}$  sono calcolati usando solo M campioni di f
- Questa trasformata è chiamata **Trasformata di Fourier Discreta**

## TRASFORMATA DI FOURIER DISCRETA INVERSA

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

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{\imath 2\pi mn/M}, \quad n = 0, \dots, M - 1$$

- Questa trasformata è chiamata **Trasformata Di Fourier Discreta inversa**

## COPPIE DI TRASFORMATE DI FOURIER DISCRETE

Le trasformate diretta e inversa di Fourier sono rappresentate come:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-\imath 2\pi ux/M}, \quad u = 0, \dots, M - 1$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{\imath 2\pi ux/M}, \quad x = 0, \dots, M - 1$$

Sono funzioni periodiche nonostante la funzione di partenza potrebbe non esserlo.

## CONVOLUZIONE CIRCOLARE

Sostituisco gli integrali con le sommatorie.

La convoluzione è detta circolare perché la convoluzione è periodica e gli M campioni sono presi all'interno di un periodo e si ripetono.

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

- La periodicità di  $g$  deriva dalla periodicità di  $f$  e  $h$
  - Questa operazione è chiamata convoluzione circolare
  - Attraverso questa operazione, il teorema di convoluzione (per la Trasformata di Fourier continua) può essere esteso alla DFT
- 

## RISOLUZIONE SPAZIALE E DELLE FREQUENZE

Qual è la relazione tra il numero di campioni presi nel dominio originale rispetto alle frequenze del dominio delle frequenze?

Gli M campioni li prendo ogni *deltaT passi*. La sequenza di campioni che prendo complessivamente è *deltaT \* numero di campioni presi*, nell'intervallo del dominio originale.

Nel dominio della frequenza sono a pezzi

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

di:

- Quindi, la DFT è definita su un intervallo di frequenze attraversato dalle sue M componenti

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

- **La risoluzione della frequenza** dipende dalla lunghezza dell'intervallo campionato nel dominio dello spazio,  $T$
- **Il range delle frequenze** coperte dalla DFT dipende dal passo di campionamento,  $\Delta T$

## Lez 10 → Trasformata di Fourier per le immagini

Abbiamo visto come passare dal dominio dello spazio o tempo al dominio delle frequenze abbiamo necessità di usare la trasformata di *Fourier*.

Fino ad ora ci siamo posti nell'ambito di una funzione a 1 variabile per semplificare le rappresentazioni.

Ci trasferiamo ora al 2D, poiché le immagini sono funzioni di 2 variabili. La parte matematica è esattamente equivalente mentre l'*aliasing* legato all'immagine è differente.

Il primo passo è quindi estendere i concetti al caso 2D:

- impulso non sarà funzione di 1 var ma di 2
  - 
  - 
  - campionamento
- Il concetto di  
La convoluzione  
Trasformata 2D  
Teorema di

### IMPULSO

La distribuzione di Dirac, o impulso, è così definita:

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

e

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

Se calcolo l'area dell'impulso deve corrispondere a 1.

Vale anche la proprietà di *sifting*: se moltiplico una funzione di 2 variabili con l'impulso centrato, l'integrale doppio è uguale al punto centrale.

La definizione di impulso nel caso discreto vale 1 nell'origine e 0 in qualsiasi altro posto:

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

---

### COPPIE DI TRASFORMATE DI FOURIER CONTINUE 2D

Nel caso continuo, devo aggiungere una 2 variabile e avrò una coppia di frequenze lungo i due assi, con un integrale doppio perché devo integrare rispetto alle variabili t e z.

La trasformata di una F di 2 variabili è uguale a:

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

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

T e z spariscono e resta la funzione in termini delle 2 componenti di frequenza. Vale anche per la trasformata inversa [seconda formula in figura].

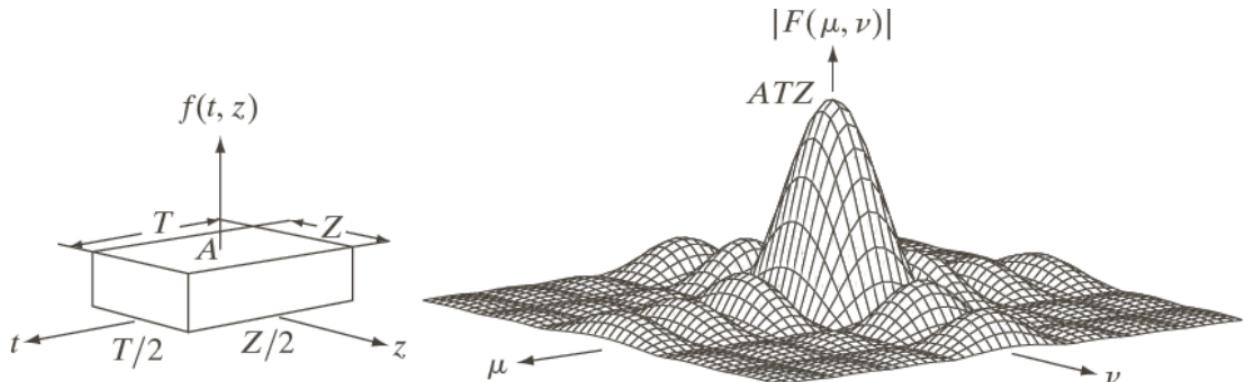
Quando abbiamo introdotto il concetto di trasformata abbiamo visto la funzione *box* e abbiamo visto quanto vale la sua trasformata. Facciamo lo stesso anche in questo caso:

- La funzione box è definita in 2D, sarà definita per un intervallo lungo l'asse t e z e la sua trasformata è una funzione *sync* 2D. intorno all'origine ho il “grosso” e poi ho tutte le piccole frequenze che si espandono.

La formula è una generalizzazione del caso 1D:

$$ATZ \left[ \frac{\sin(\pi\nu T)}{\pi\nu T} \right] \left[ \frac{\sin(\pi\mu Z)}{\pi\mu Z} \right]$$

Più la funzione box è ampia, più la funzione sync è stretta. Questo discorso si applica componente per componente.



## TEOREMA DI CAMPIONAMENTO 2D

Abbiamo visto che se ho una funzione di dominio nello spazio F in 1 var per campionarla devo moltiplicarla per un treno di impulsi, faccio lo stesso anche per il 2D ma anche il treno di impulso ha 2 variabili. Moltiplico e ottengo un'espressione legata al treno lungo i due assi.

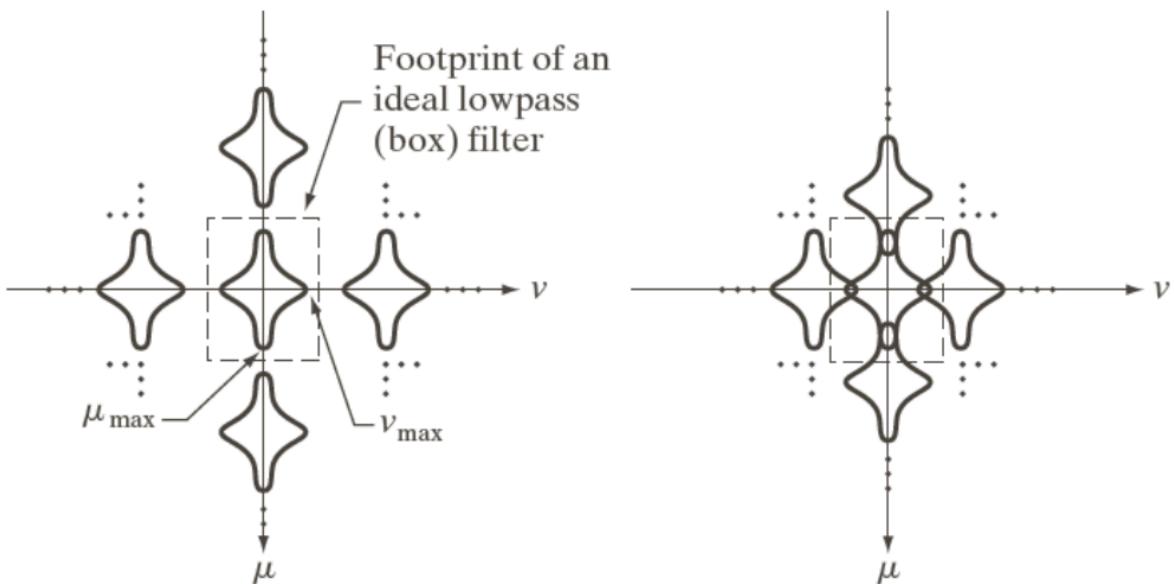
Il teorema: il tasso di Nyquist deve essere rispettato lungo le due variabili di frequenza.

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

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

## ALIASING NELLE IMMAGINI

Se non rispettiamo il tasso di nyquist in una delle 2 dimensioni o entrambe, si introduce *aliasing* perché la trasformata della funzione campionata è una funzione periodica dove il singolo periodo è la trasformata della funzione originale, se non rispettiamo il tasso non riusciamo a isolare il singolo periodo e risalire a esso per applicare la trasformazione e ricostruire il segnale.



[in figura vediamo i periodi accavallati, è l'*aliasing*].

L'aliasing in un'immagine lo si vede quando ci sono pattern periodici: le linee appaiono frastagliate oppure all'interno dell'immagine si possono creare artefatti *ex novo* che non hanno corrispondenza con l'immagine originale.

Il teorema di nyquist consente di non avere aliasing ma nella pratica c'è sempre perché tutte le funzioni reali per i nostri scopi sono *limitate nel tempo*.

In virtù di ciò, prima di discretizzare un segnale, posso fare uno *smoothing* applicando un filtro in modo da ridurre le frequenze alte, se faccio così l'aliasing introdotto dalla funzione limitata viene limitata.

## RICAMPIONAMENTO E INTERPOLAZIONE

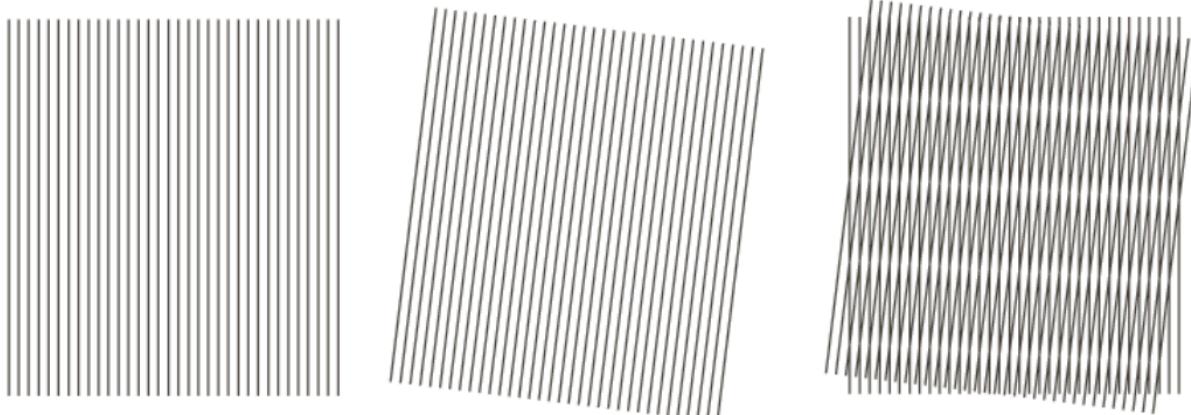
L'aliasing si introduce anche quando lavoro con l'immagine digitale e le manipolo, in particolare con zoom in/out [se faccio ridimensionamento faccio un sottocampionamento, se faccio sovradimensionamento faccio un sovraccampionamento] ma in particolare quando riduco un'immagine: questa operazione introduce *aliasing* perché introduce pattern di frequenza non contenuti nell'immagine originale.

In questo caso è possibile effettuare un *de aliasing* facendo uno smoothing dell'immagine, la differenza rispetto a prima è che non lavoro sul segnale ma su un'immagine già digitalizzata e fare in modo che gli artefatti introdotti siano contenuti.

## EFFETTI MOIRE'

Altri artefatti introdotti nelle immagini, non per forza legati all'*aliasing*, è *l'effetto moiré*.

È quando ho dei pattern periodici simili che vado a sovrapporre si crea un effetto a "zanzariera".



Questo effetto si genera anche nelle stampe di foto: la foto viene campionata orizzontale e verticale ma nella stampa vengono usati dei *semitoni* per replicare l'intensità dei pixel. Questi semitonni hanno un orientamento di 45° gradi e non allineata con l'immagine campionata, questa differenza si manifesta sottoforma di effetto moiré.

Applicando particolari filtri possiamo attenuare ma zoomando vediamo ancora degli artefatti.

---

### COPPIA DI FOURIER BIDIMENSIONALE [DFT]

Il prossimo passo da fare è trasformare Fourier da discreta a 2D.

Abbiamo M punti lungo un'asse e N punti lungo l'altro asse

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-\imath 2\pi(ux/M + vy/N)}$$

### PROPRIETA' DELLA DFT

Ci sono proprietà importanti per la costruzione di opportuni filtri.

1. Traslazione: se prendiamo la funzione originale e la moltiplichiamo per il valore dell'esponenziale, equivale traslare la trasformata dall'origine a  $u_o$  e  $v_o$ . vale anche il contrario: se prendo la trasformata centrata in 0,0 e la moltiplico per l'esponenziale equivale a traslare la funzione originale dall'origine a  $x_o$   $y_o$ .
  2. Rotazione: la si vede esprimendo le trasformate in coordinate polari. Se ruoto l'immagine di partenza di un certo angolo, anche la trasformata sarà ruotata dello stesso angolo.
  3. Periodicità.
  4. Simmetria.
- 

### SPETTRO DI FOURIER E ANGOLO DI FASE

Se esprimo in forma polare la trasformata di *Fourier* posso calcolare lo spettro di Fourier che è dato dalla magnitudine delle componenti delle frequenze delle immagini.

Se ho una funzione complessa, lo spettro è calcolata come la radice quadrata delle componenti reale e quella immaginaria, elevate al quadrato e poi sommate.

L'angolo di fase è l'arcotangente del rapporto tra la parte immaginaria e reale. Misura quanto le componenti di frequenza del segnale sono traslate rispetto all'origine.

Sono tutte matrici  $m \times n$  e possono essere visualizzate.

*La componente centrale dello spettro di Fourier ha un valore di magnitudine più elevata, detta anche dc.*

---

### SPETTRO DI FOURIER

Rappresenta l'ampiezza delle frequenze contenute nell'immagine. Più l'ampiezza è elevata, più la frequenza è predominante nell'immagine e viceversa.

Lo spettro di *Fourier* è sensibile alla rotazione ma non alla traslazione.

### ANGOLO DI FASE

Se lo rappresentiamo graficamente non ci dà nessuna info particolare, ma è fondamentale poiché porta dentro di sé un'info fondamentale: ci dice quanto le *sinusoidi* si discostano rispetto all'origine. Siccome le ***sinusoidi*** sono associati agli oggetti dell'immagine, l'angolo di fase ci dice la posizione degli oggetti nell'immagine.

---

### TEOREMA DI CONVOLUZIONE 2D

Se considero la trasformata della convoluzione di 2 immagini equivale al prodotto del dominio delle frequenze delle trasformate.

Abbiamo definito la convoluzione circolare perché le funzioni sono periodiche. Con queste due equazioni vediamo l'equivalenza tra il filtraggio del dominio spaziale e del dominio delle frequenze: o filtriamo in un dominio o nell'altro otteniamo lo stesso risultato però sotto quali condizioni le due operazioni sono equivalenti?

Quando consideriamo la trasformata discreta e facciamo convoluzione dobbiamo tener conto della periodicità.

La convoluzione equivale a far scorrere la seconda funzione sulla prima e calcolare il prodotto ma dobbiamo prima ribaltare la funzione  $h$  di  $180^\circ$ . quando faccio la convoluzione ottengo un *trapezio*, se erano due funzioni *box*.

Abbiamo detto che quando trasformiamo con trasformate discrete, la convoluzione è circolare, cioè periodica. Quando considero l'operazione precedente, la funzione  $f$  si può estendere all'infinito e si ripete, lo stesso vale per  $h$ . quando faccio convoluzione c'è un problema: agli estremi i periodi successivi interferiscono con la convoluzione e avremo un risultato differente.

Come si fa a risolvere?

Si ottiene un errore di WRAPAROUND che è dovuto alla convoluzione circolare.

Per evitare questo errore si usa lo zero padding: se ho due sequenze di campioni A,B per evitare l'errore posso aggiungere degli zeri in un numero tale per cui ottengo una nuova sequenza P tale che  $P \geq A + B - 1$ . In questo modo, la convoluzione spaziale e la circolare forniscono stesso risultato.

Nel caso delle immagini, ipotizziamo  $AxB$  e  $CxD$ : faccio lo zero padding in modo da avere una nuova DIM  $PxQ$ .

Se una funzione non è zero alla fine dell'intervallo, lo zero padding introduce artefatti (*effetti blocco*).

*In generale moltiplico per attenuare l'errore, prendiamo la funzione considerata per una gaussiana che ha gli estremi tendenti a zero, e poi si fa zero padding.*

## Lez 11 → Filtraggio nel dominio delle frequenze

Abbiamo visto come i concetti della trasformata di Fourier si estendono dal caso 1D al 2D e abbiamo visto nello specifico alcune questione legate al campionamento e all'aliasing sulle immagini.

Quando si lavora nel dom della frequenza, ragionare in termini di operazioni di filtraggio è più semplice perché c'è una corrispondenza diretta tra le caratteristiche delle immagini e il contenuto in frequenza: quando ci sono variazioni di intensità elevate, corrispondono ad alte frequenze. Siccome i picchi di frequenza sono ai **lati**, se vogliamo fare **sharpening** dobbiamo preservare le frequenze alte e tagliare le basse, se vogliamo eliminare il rumore dobbiamo mantenere le basse e abbattere le alte, sfocando quindi l'immagine. Vedremo dei filtri *passa basso e passa alto*.

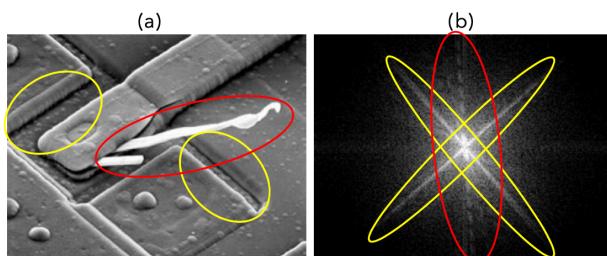
### ANALISI BASATA SULLA TRASFORMATA DI FOURIER

Se prendo un'immagine nel dom spaziale e determino la TF (trasformata di Fourier) e visualizzo il suo spettro, abbiamo la possibilità di ragionare sullo spettro e collegare alcuni suoi aspetti sull'immagine vista nel dominio dello spazio?

Se prendo un'immagine e ne faccio la trasformata ottengo la trasformata 2D discreta, **possiamo fare un collegamento tra la trasformata e sul contenuto dell'immagine nel dom delle frequenze?** Se fisso 2 frequenze nel dominio della trasformata, fissato un valore nella trasformata, il valore dipende da tutti i valori dell'intera immagine. Se ho una coppia di frequenze nel dominio della trasformata non c'è un collegamento diretto. Quello che possiamo fare sono considerazioni generali:

- Se ho porzioni di immagini in cui ci sono oggetti, certamente corrispondono a frequenze elevate nella trasformata, e viceversa.
- Possiamo analizzare lo spettro per avere info importanti per mettere su dei filtri nel dom delle freq.

### ANALISI BASATA SULLA TRASFORMATA DI FOURIER



Nell'immagine di partenza ci sono alcune demarcazioni che sono molto evidenti e sono disposti a circa  $45^\circ$ , se vediamo lo spettro infatti vediamo componenti a  $45^\circ$ ...

---

## FILTRAGGIO NEL DOMINIO DELLE FREQUENZE

In generale, possiamo lavorare in un dominio o nell'altro, perché scegliere uno invece che un altro? Ci sono ragioni di natura computazionale: se devo lavorare con immagini e filtri di piccole dimensioni, conviene lavorare nel dominio spaziale, tuttavia, per applicazioni specifiche con filtri di dimensioni maggiore, conviene più usare il dominio delle frequenze perché la TF viene calcolata con la TF Veloce ed è più conveniente.

Se lavoro nel dom delle frequenze devo fare la trasformata dell'immagine e devo modificarla applicando un filtro. Fatto ciò, torno nel dominio spaziale e visualizzo il risultato ottenuto.

Applico la formula, concepisco un filtro nel dom delle freq, lo moltiplico per la trasformata e vado ad applicare la Trasformata inversa dove otterrò una nuova immagine che è il risultato del filtraggio.

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

- $H(u, v)$  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

---

## FILTRO NEL DOMINIO DELLA TRASFORMATA

- Possiamo costruire un semplice filtro  $H(u, v)$ 
  - $H(u, v) = 0$ , se  $u = v = 0$ ,
  - $H(u, v) = 1$ , altrimenti
- Impostando a zero solo la componente dc ha l'effetto di spostare a zero la media di  $g$ 
  - Se i livelli di grigio non sono scalati, **g sarà più scura**

## FILTRI PASSA BASSO E PASSA ALTO

Posso costruire dei filtri che azzerano (tagliano) un range di frequenze, le frequenze elevate sono dettagli e le basse sono background ecc...  
Posso applicare filtri per fare *sharpening e smoothing*.

- ***Un filtro passa basso*** fa passare le frequenze più basse, elimina le alte e i dettagli e sfoca.
  - ***Un filtro passa alto*** elimina le frequenze basse, esalta i dettagli ma perdiamo anche la frequenza *dc* (0,0) e abbasso il contrasto.
- 

## WRAP AROUND E PADDING

L'errore di *wrap around* è introdotto dal fatto che le trasformate discrete sono funzioni periodiche e la convoluzione sarà circolare e con la convoluzione circolare, quando vado a fare l'operazione di convoluzione, il fatto che le trasformate sono periodiche, i periodi adiacenti influenzano il calcolo dell'operazione e rispetto alla versione non periodica c'è una differenza che è questo errore, per evitare ciò bisogna fare zero padding per separare i due periodi che si ripetono.

Se faccio il padding dell'immagine originale, evito l'errore di WA. Il problema è che il padding lo faccio nel dominio spaziale ma il filtro è definito nel dominio delle frequenze!

In linea teorica dovrei applicare il filtro anche nel dominio delle frequenze, come posso risolvere?

- Definiamo il filtro, mi calcolo l'anti trasformata del filtro, vado nel dom spaziale, faccio padding e ritorno nel dominio delle frequenze facendo la trasformata. Non possiamo fare ciò perché si introduce un altro fenomeno, detto *ringing*.

In definitiva, non facciamo il padding della rappresentazione spaziale del filtro ma facciamo solo il padding dell'immagine originale. Il dom della freq lo usiamo per definire il filtro le cui dimensioni sono pari a quelle dell'immagine col padding.

Una forma di WA ci sarà sempre l'effetto introdotto è sia limitato dal padding, ed è anche meglio del *ringing*.

---

## FILTRI DI SHIFT ZERO PHASE

Tutti i filtri che vedremo sono *filtri shift zero phase* perché quando faccio filtraggio, l'angolo di fase viene preservato perché perdiamo il contenuto informativo dell'immagine se non lo avessimo.

## PROCEDURA DI FILTRAGGIO DELLE FREQUENZE

Ho un'immagine, calcolo P e Q per fare padding e lo faccio.

Vado a centrare l'immagine. Calcolo la trasformata dell'immagine *padded*, creo un filtro che conviene sia simmetrico e reale così l'anti trasformata sarà reale, calcolo la trasformata filtrata moltiplicando la trasformata per il filtro, vado a invertire il precedente prodotto, ottenendo l'immagine nel dom spaziale padded e faccio *cropping* per ripristinare le dimensioni originarie.

## RISPOSTA DI IMPULSO

Questi filtri sono anche detti filtri a risposta di impulso finiti (FIR).

---

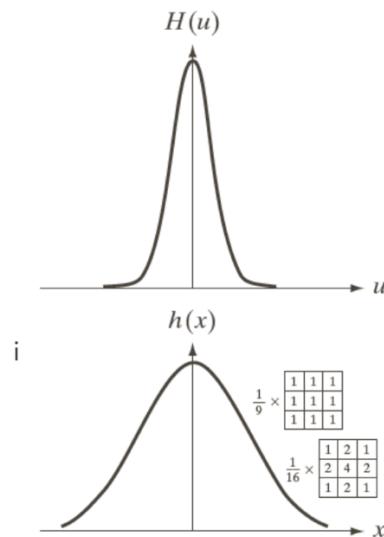
## PROGETTAZIONE DI FILTRI SPAZIALI

Si usa il dom delle frequenze per progettare filtri spaziali.

Supponiamo di avere nel dom delle frequenze un filtro *gaussiano (passa basso)*, se calcolo l'anti trasformata ottengo un'altra gaussiana.

Osserviamo che i valori sempre positivi e che verso il centro devo avere valori elevati.

Se definisco un filtro nel dom delle frequenze, devo farlo con le dimensioni dell'immagine padded e quindi ho un filtro molto più grande che viene usato per creare piccole maschere.



Posso progettare anche filtri nel dom spaziale, nel dominio delle freq posso considerare un filtro dato dalla differenza di due gaussiane. L'effetto della differenza è una funzione che corrisponde a un filtro *passa alto*.

---

## FILTRI PASSA BASSO

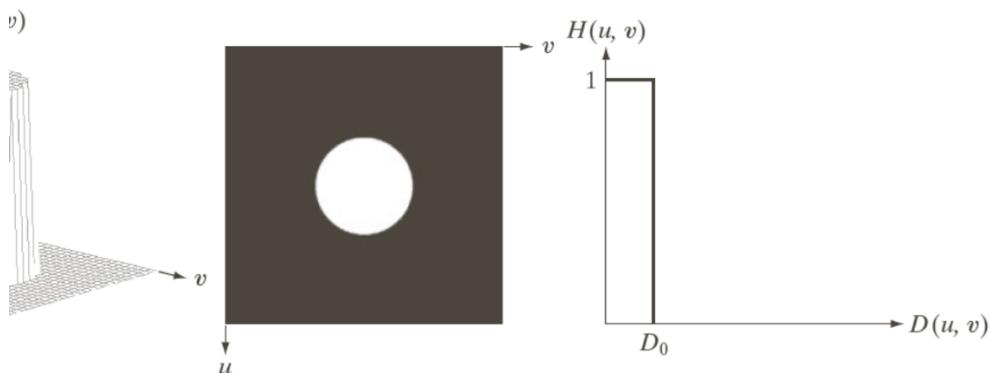
Ne vediamo 3:

- 1.Filtro passa basso ideale: fissato il centro dello spazio delle frequenze, taglio tutte le frequenze oltre un cerchio di raggio  $D_0$  (frequenza di *cut off*). Per tutte le frequenze nel cerchio vale 1, oltre vale zero.
- 2.Filtro di *Butterworth*: ci consente di avere un comportamento che si avvicina o al filtro ideale o al gaussiano.
- 3.Filtro Gaussiano.

## FILTRO IDEALE PASSA BASSO

Tutte le frequenze nel cerchio le conserviamo e all'esterno le tagliamo.

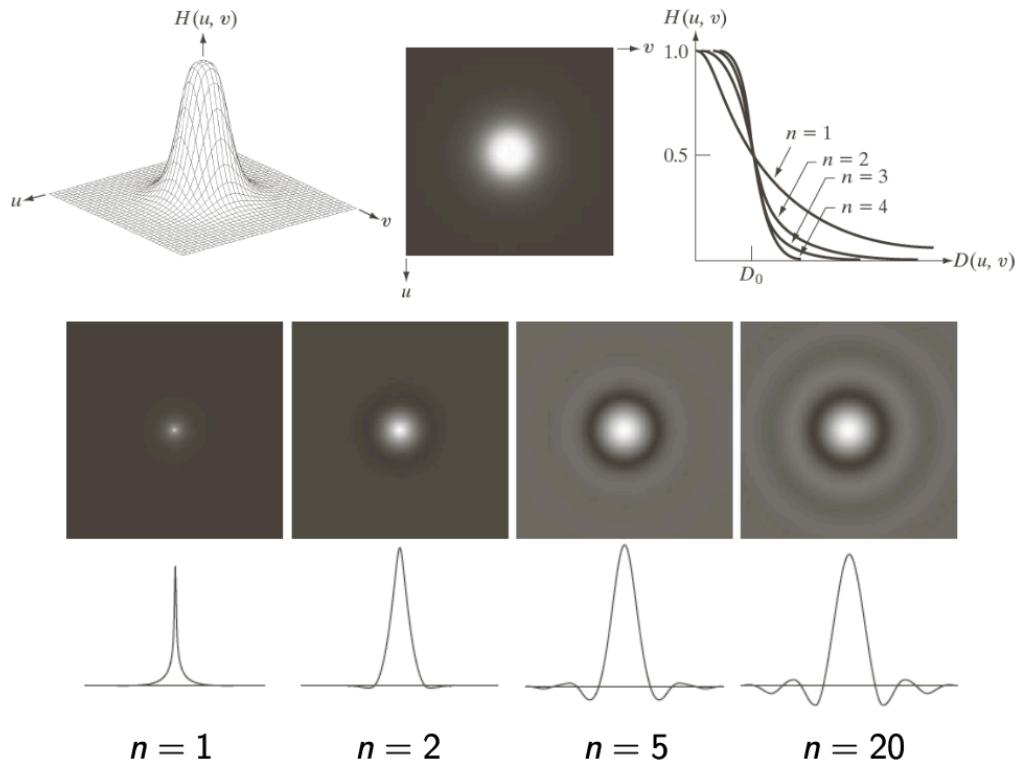
$$H(u, v) = \begin{cases} 1, & D(u, v) \leq D_0 \\ 0, & D(u, v) > D_0 \end{cases}$$



Per scegliere il *cut off* si determina la potenza contenuta all'interno del cerchio.

## FILTRO PASSA BASSO DI BUTTERWORTH

Fisso una frequenza di *cut off* e  $N$  ne determina l'aspetto. Per valori di  $N$  crescenti, l'aspetto della funzione comincia a diventare simile al filtro ideale, più grande è  $N$  più sono al filtro ideale, più piccolo mi avvicino alla Gaussiana.



## FILTO PASSA BASSO GAUSSIANO

Rispetto al filtro di ***Butterworth*** consente uno smoothing più graduale. Siccome l'anti trasformata di una Gaussiana è una Gaussiana, non avremo mai problemi di *ringing*.

## FILTRI PASSA ALTO

Sono ottenuti dai passa basso facendo: ***1 - passa basso.***

**Sono sempre i 3 visti nei passa basso.**

## FILTRO IDEALE PASSA ALTO

Per valori minori della frequenza di *cut off* tagliamo e per maggiori passano.

## LAPLACIANO (*si pronuncia laplasiano*)

Abbiamo visto come calcolare il laplaciano e dopo lo si somma all'immagine originale per avere un'immagine migliore.

Nel dominio delle freq si può dimostrare che il laplasiano è

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

A prescindere dal fatto che il laplasiano sia centrato in 0,0 o meno, dobbiamo calcolare il laplaciano e farlo nel dominio della frequenza, usando tale equazione:

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

*Trasformata di Fourier dell'immagine originale \* laplaciano*

L'immagine filtrata ottenuta con:

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

Nel dominio spaziale, i valori delle intensità dell'immagine originale  $f(x,y)$ , sono paragonabili al laplaciano, nel dominio delle frequenze invece le operazioni portano a una magnitudine più elevata rispetto all'immagine originale.

Per far sì che il laplaciano abbia dei valori simili all'immagine originale si calcola il laplaciano e lo si normalizza e viene normalizzata anche l'immagine originale prima di calcolare la trasformata di Fourier, avendo così valori tra -1 e 1.

Nel dominio spaziale applichiamo una maschera pezzo per pezzo in regioni locali, nel dominio delle freq invece lavoriamo sulle intere immagini, ecco perché gli output sono migliori.

---

## MASCHERAMENTO DI UNSHARP

Dal dominio dello spazio sappiamo che se ho un'immagine e determino una versione sfocata (**blurred**), questa versione la posso sottrarre all'immagine

originale, il risultato è  $g_{mask}$ . Questa maschera la posso aggiungere all'immagine originale per avere un **mascheramento di unsharp** che mette in evidenza i dettagli.

Nel dominio spaziale può essere contestualizzato così:

la maschera la otteniamo come sottrazione tra immagine originale e versione smoothed ( $f_{lp}$  [filtro passa basso applicato]). Il filtro  $f_{lp}$  la ottengo come anti trasformata del prodotto tra la trasformata di F e il filtro passa basso (di solito gaussiano per evitare *ringing*).

*K* se è 1 è la unsharp mask mentre se è >1 otteniamo un **filtraggio highboosting**.

- La tecnica di mascheramento di unsharp richiede una maschera  $g_{\text{mask}}$

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

- dove

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

- L'immagine filtrata è

$$g(x, y) = f(x, y) + k g_{\text{mask}}(x, y)$$

$k = 1$  unsharp masking  
 $k > 1$  highboosting

- Nel dominio della frequenza

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

$1 - H_{\text{LP}}$  è semplicemente un filtro passa alto.

Quando applichiamo un passa alto ci azzera la frequenza di  $c$ , il problema qui non si pone perché non perdo info sulle intensità perché al filtro aggiungo un 1, preservando le frequenze.

Infatti:

- Può essere espresso in termini di filtro passa alto:

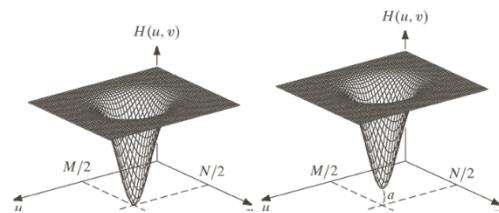
$$g(x, y) = \mathcal{F}^{-1}\{(1 + k H_{\text{HP}}(u, v)) F(u, v)\}$$

- Generalizzando

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

- Dove

- $k_1 \geq 0$  è l'offset dall'origine
- $k_2 \geq 0$  è il contributo delle alte frequenze



## FILTRAGGIO OMOMORFICO

Quando abbiamo illustrato i concetti di immagine, abbiamo detto che un'immagine è rappresentabile come il prodotto di due componenti: illuminazione e riflettanza (quantità di luce che l'oggetto riflette).

Possiamo esprimere l'immagine come prodotto di queste componenti.

Se volessi fare così nel dominio delle freq e operare sulle due componenti separatamente non posso farlo direttamente: se calcolo la trasformata di  $F$

dell'immagine, è diverso da fare il prodotto delle trasformate. Non sono in grado di operare sulle singole componenti  $i, r$ .

Per risolvere si adotta un trucco: si esprime l'immagine in una versione logaritmica. Definisco una nuova immagine che è il log di  $f(x, y)$  che è  $i(x, y) * r(x, y)$ .

Il prodotto di due log è la somma di due log e posso ora applicare la trasformata di F.

- Il logaritmo ha l'interessante proprietà

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

- Per la linearità della DFT

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

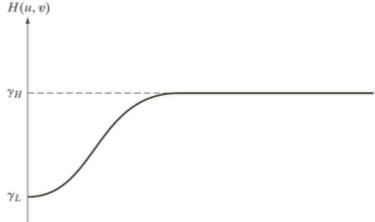
- Il filtro H può essere applicato ad ambo le componenti

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

- Nel dominio spaziale, la trasformazione logaritmica può essere invertita

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

- Il processo può essere realizzato usando un filtro come



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

A questo punto applico il filtro nel dominio delle freq.

Quando definiamo il filtro,  $H$ , lo applichiamo alla trasformata, cioè alle due componenti della trasformata, in sostanza riesco a operare un filtraggio separato sulle due componenti.

Viene fatto ciò perché la componente di illuminazione è associata a piccole variazioni di frequenza mentre la componente di riflettanza a variazioni repentine. Sapendo ciò possiamo migliorare l'immagine usando un filtro che controlli le due componenti, usando un filtro adeguato (mix tra passa basso e passa alto).

## FILTRI BAND-REJECT E PASSA BANDA

Sono filtri **selettivi** perché lavoriamo nel dom delle freq e l'idea è che mentre coi passa bassa/alto lavoriamo su tutte le frequenze, **con questi possiamo manipolare un range di frequenze delle immagine**.

Ci sono 2 categorie di filtri selettivi:

-Definisco un intervallo e decidere se farle passare o meno, sono i filtri passa banda o *band reject* a seconda se all'interno dell'intervallo le facciamo passare o le tagliamo.

-Si opera su un'area di frequenza, filtri *notch*.

Per quanto riguarda i band reject, dopo averli definiti, ottengo il passa banda facendo **1 - filtro band reject**.

Posso definire i band reject che sono versioni modificate del passa basso ideale, gaussiano e Butterworth.

E' possibile ottenere il filtro band-pass dal band reject nel seguente modo:

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

## FILTRI NOTCH

È un filtro selettivo. Notch significa buco.

**Si va a lavorare su un'area.** Possono decidere che in quell'area le frequenze devono essere abbattute o passare.

Si usa il notch per rimuovere del rumore periodico che si trova nello spettro di Fourier.

Abbiamo detto che sono filtri che non alternano l'angolo di fase. Quando definisco un notch deve essere anche **simmetrico** perché se così non fosse avremmo che l'immagine nel dom spaziale dovrebbe essere **complessa**.

Andiamo a definire quindi un filtro, nel dom delle freq, che è centrato in un punto che è l'area da manipolare. Questo punto è il punto di notch e lo si sceglie lavorando sullo **spettro** di F. Posso definire un filtro di notch di rigetto (nell'area le frequenze devono essere eliminate) facendo il prodotto di filtri passa alto centrati nel punto di notch e il suo simmetrico. Se ho 3 componenti periodiche associate al rumore, avrò 3 prodotti di un filtro di notch centrato e un filtro di notch centrato nel suo simmetrico.

- Poiché i filtri zero-phase shift sono simmetrici rispetto al centro del rettangolo della frequenza, un filtro notch con centro  $(u_0, v_0)$ , deve avere una corrispondente punto notch in  $(-u_0, -v_0)$
- Un filtro notch di rigetto è costruito come prodotto di filtri passa alto i cui 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

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

- Esempio

- $H_{NR}(u, v) = \prod_{k=1}^3 \left[ \frac{1}{1 + [D_{0k}/D_k(u, v)]^n} \right] \left[ \frac{1}{1 + [D_{0k}/D_{-k}(u, v)]^n} \right]$

- Un filtro notch passante è ottenuto da un filtro notch di rigetto mediante
    - $H_{NP}(u, v) = 1 - H_{NR}(u, v)$
- 

## COMPUTAZIONE DELLA DFT

Abbiamo visto la corrispondenza tra dominio spaziale e delle frequenze, grazie al teorema di convoluzione che lega i due mondi.

### Dove conviene lavorare?

Abbiamo detto che se dobbiamo manipolare un'immagine è più facile lavorare nel dominio delle frequenze perché c'è una corrispondenza diretta con l'output desiderato. Però in generale lavorare nel dominio spaziale è meno costoso computazionalmente, facendo però dei ragionamenti:

-Quando calcolo la trasformata di F discreta, è **separabile**: se ho una trasformata 2D la posso ottenere come il prodotto di 1D che è un risparmio computazionale.

-La DFT è stata migliorata, la **FFT (fast Fourier)** dove i tempi computazionali sono abbattuti, da una componente quadratica a una **logaritmica**.

Questo però non risponde alla domanda. Dipende:

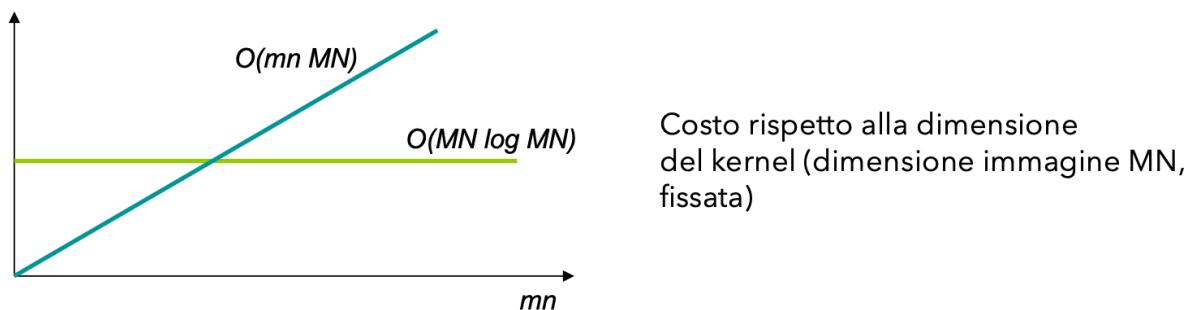
-Il costo della FFT è logaritmico quindi quando faccio un filtraggio nelle frequenze, la trasformata di F dipende solo dalle DIM dell'immagine e non del kernel.

-Nel dominio spaziale la complessità è dimensione del kernel che moltiplica MN. C'è un punto sotto il quale conviene lavorare nel dominio spaziale ma oltre ad esso il costo computazionale della convoluzione del dom spaziale supera quello del dominio delle frequenze.

### Concludendo:

**Quando abbiamo immagine con kernel piccolo conviene il dom spaziale, se ci sono applicazioni avanzate con kernel grandi, conviene il dom delle frequenze.**

- Il costo del filtraggio è dominato dal calcolo della FFT
- Il costo della convoluzione nel dominio spaziale è  $O((mn)(MN))$ , con mn dimensione del kernel
- La convoluzione nel dominio della frequenza è molto più veloce per kernel grandi



## Lez 12-> MORFOLOGIA APPLICATA ALLE IMMAGINI

La morfologia matematica riguarda l'applicazione di operatori morfologia all'elim.

Per morfologia intendiamo la forma degli esseri viventi.

Nel caso dell'elim si intendono una serie di operatori che lavorano sulla forma degli oggetti contenuti nell'immagine per identificare regioni che descrivono l'oggetto come bordi o altre caratteristiche salienti di oggetti.

Possiamo dire che la morfologia matematica per l'elim descrive le forme rappresentate nelle immagini.

Gli oggetti come li possiamo trattare per applicare gli operatori?

Li trattiamo come caratterizzati da un insieme di pixel che sono distribuiti nel piano a seconda di come gli oggetti sono distribuiti.

La cosa importante è che quando si lavora con la morfologia, si sfrutta la conoscenza a priori degli oggetti che si trattano per lavorare in modo da individuare le specifiche caratteristiche dell'oggetto in questione.

Quando diciamo che gli oggetti sono insieme di pixel, quando lavoriamo con morfologia, lavoriamo con gli *insiemi*. I pixel rappresentano o gli oggetti (*fg*) o lo sfondo (*bg*).

I processi morfologici sono operazioni su insiemi e i nostri insiemi sono punti del piano e le loro coordinate sono intere (x,y) e quindi sono punti dello spazio Z2 dove Z è l'insieme degli interi.

Gli insiemi fanno riferimento agli oggetti contenuti nelle immagini. Significa che questo concetto è applicabile alle immagini binarie dove i pixel sono 0 o 1. Se ho un'immagine binaria, rappresentare l'insieme è molto semplice perché considero come insieme tutti quei pixel il cui valore è 1: assumiamo che il fg è bianco il bg col nero.

Se ho un'immagine binaria  $f$ , posso descrivere un insieme  $B$  come un sottoinsieme di Z2 così:

se  $f(x,y) = 1$  appartiene a  $B$ , se è 0 non appartiene a  $B$ .

$B$  è l'insieme dei pixel per cui il valore del pixel è 1.

Possiamo rappresentare anche le immagini a scala di grigio, dove il vettore diventa *tridimensionale*, dove aggiungo il valore di intensità.

Usiamo due tipi di insiemi di solito:

-Gli oggetti: tutti quelli che contengono pixel bianchi.

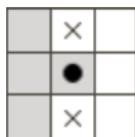
-Elemento strutturante: può essere costituito da pixel con valore 1 ma anche con valori 0. Si chiama così perché svolge un ruolo simile ai *kernel* per la convoluzione: è un insieme che uso per individuare specifiche caratteristiche morfologiche degli oggetti nell'immagine.

---

## ELEMENTI STRUTTURANTI

Le operazioni morfologiche hanno come attore principale l'elemento strutturante. Questi sono definiti rispetto a un'origine: è un array di pixel e dobbiamo decidere il centro. Gli elementi strutturanti sono definiti usando la convenzione:

- Il centro è il pallino nero.
- Le celle in grigio rappresentano gli insiemi di fg (1)
- Quelle bianche sono bg (valore 0)
- Le  $x$  sono *don't care*.



## INSIEMI E IMMAGINI

Poiché le immagini sono matrici e sono rettangolari, gli insieme non è detto che siano rettangolari. Per usare le immagini e un elemento strutturante, è importante che anch'esso sia rettangolare perché l'idea è far scorrere l'elemento sulla nostra immagine, proprio come un kernel.

Sia immagini che elementi strutturanti devono essere array di pixel.

Questo aspetto è esattamente il parallelismo tra kernel e convoluzione e immagini ed elementi strutturanti: il bg degli oggetti è tipo padding e gli elementi strutturanti come kernel che scorrono l'immagini.

---

## ELEMENTI STRUTTURANTI

Se ho elementi strutturanti, per usarli li devo prima trasformare in array di pixel.

---

## DEFINIZIONI

Gli oggetti su cui lavoriamo sono degli insiemi e quindi valgono le operazioni insiemistiche a cui si aggiungono 2:

- Riflessione: (**B hat**) è l'insieme di tutti i punti dell'insieme B riflessi, inverto il segno delle coordinate da( x,y ) a ( -x,-y ).

-Traslazione: è definito rispetto a un insieme e a un punto  $z$  ( $z_1, z_2$ ).

Traslare  $B$  rispetto a  $z$  significa considerare tutti i punti di  $B$  traslati in  $z$ : è l'insieme di tutti i punti di  $B$  traslati di  $z$ .

Queste operazioni possiamo applicarle agli elementi strutturanti.

---

## OPERATORI MORFOLOGICI

### EROSIONE

Gli operatori morfologici sono scritti in termini di elemento strutturante applicato a un oggetto di fg.

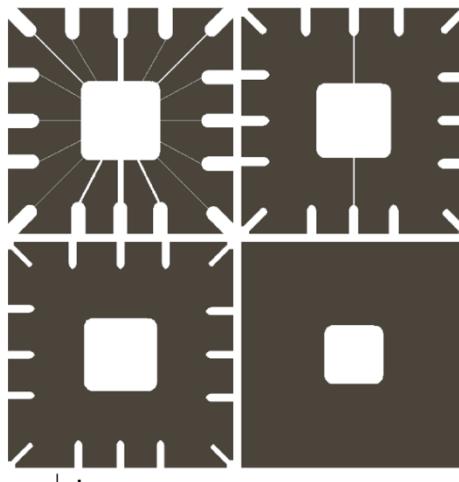
Considero  $A$  insieme di oggetti e  $B$  elemento strutturante.

Dati  $A$  e  $B$ , l'erosione di  $A$  attraverso  $B$  è definito come un insieme: è l'insieme di tutti le traslazioni dell'elemento strutturante tale che esso è contenuto all'interno dell'oggetto  $A$ .

Cosa fa l'erosione? Dipende tutto dalle dimensioni dell'elemento strutturante: l'oggetto viene rimpicciolito a seconda dello strutturante. Più l'elemento è grande, più c'è erosione.

### FILTRAGGIO MORFOLOGICO

L'erosione si può usare come una sorta di filtraggio morfologico perché tolgo tutto ciò che ha una determinata forma che abbiamo deciso. Nel caso dell'erosione posso fare un filtraggio morfologico.



Se voglio cancellare le linee basta usare un elemento strutturante con dimensione maggiore delle linee che vogliamo eliminare.

Concludendo, l'erosione cancella i dettagli dell'immagine più piccoli dell'elemento strutturante.

---

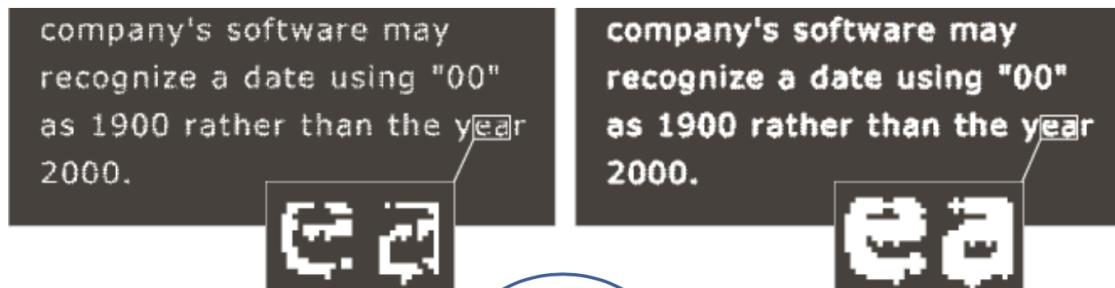
### DILATAZIONE

La dilatazione è definita anche in termine di riflessione.

Se ho insiemi A e B (oggetto e strutturante), la dilatazione di A attraverso B è definita come tutte le traslazioni dell'insieme B riflesso, tale per cui intersecato con l'oggetto abbia almeno un elemento in comune.

Questo significa che innanzitutto devo ribaltare B, poi faccio le traslazioni e le metto insieme dilatato a patto che se intersecate con l'oggetto, abbiano un elemento in comune almeno, ecco perché le immagini si dilatano (basta anche un solo elemento in comune).

Un'applicazione della dilatazione è quella riguardante il miglioramento del testo per riempire le lettere interrotte.



La dilatazione è simile al filtraggio passa basso, i dettagli sono assorbiti.

---

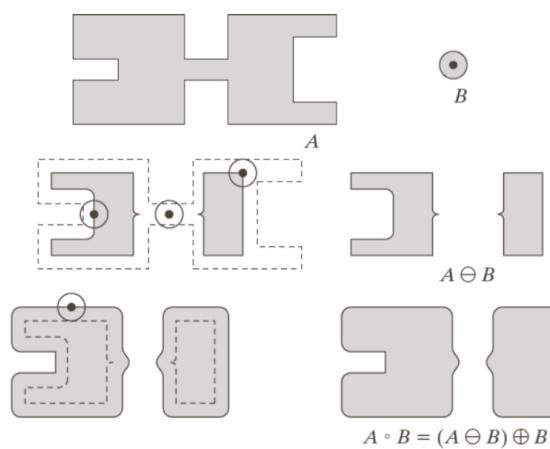
## APERTURA e CHIUSURA

Sono altri 2 operatori.

### APERTURA

L'aperura di A attraverso B è definita come: prendo A, l'erofo con B e il risultato lo vado a dilatare sempre con B.

L'effetto è rimuovere regioni che non sono in grado di contenere l'elemento strutturante e rende omogeneo i contorni dell'oggetto ed elimina ponti e protuberanze.

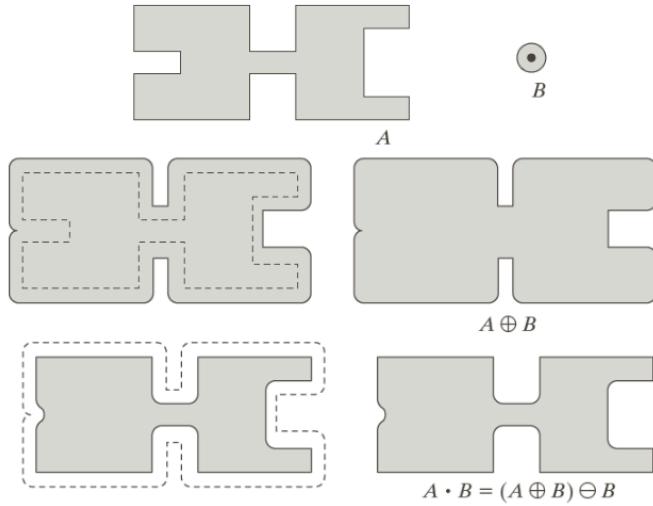



---

## CHIUSURA

Prima dilato A e il risultato lo erofo con B.

Anche qui i contorni sono omogenei ma se ci sono piccole interruzioni diventano fuse. Le zone concave le va a riempire.



## APERTURA E CHIUSURA

Possiamo interpretare geometricamente apertura e chiusura:

L'apertura rappresenta i punti di A coperti dalla traslazione di B lungo il bordo interno di A.

La chiusura aggiunge ad A quei punti del bg che non sono coperti dalla traslazione di B lungo il bordo esterno di A.

Anche apertura e chiusura godono della dualità rispetto a complemento e riflessione.

Concretamente possono essere usate per fare filtraggio del rumore.

## TRASFORMAZIONI HIT OR MISS

Sono usate per trovare oggetti diversi in un'immagine.

Per fare questo, gli oggetti devono essere separati da almeno 1 pixel di bg.

L'elaborazione Hit or Miss si basa su 2 elementi strutturanti:

$B_1$  è la forma dell'oggetto che voglio individuare nell'immagine.

$B_2$  serve per cercare le forme nel bg.

Quindi un elemento si applica agli oggetti e un altro al bg.

La trasformazione Hit or miss: data l'immagine ne faccio la trasformazione hit or miss rispetto a  $B_1$  e  $B_2$  ed è un insieme di tutte le traslazioni  $z$  per cui la

traslazione  $B_1$  è inclusa nell'oggetto all'interno dell'immagine e la traslazione di  $B_2$  deve essere inclusa nel complemento di A (foreground).

Quando applico  $B_2$  al complemento, cerchiamo un *hit* che corrisponde a un *miss* nel foreground originale.

---

## MORFOLOGIA IN SCALA DI GRIGIO

Abbiamo visto tutto per le immagini binarie, ma possiamo farlo anche in scala di grigio, modificando gli operatori.

Innanzitutto non posso considerare come base delle operazioni morfologiche gli insiemi, poiché abbiamo coordinate e intensità: gli oggetti e gli elementi strutturanti saranno ora funzioni.

Quando si lavora in scala di grigio, l'elemento strutturante può essere *flat* e *non flat*: nel *non flat* abbiamo valori di intensità variabili mentre nel *flat* si definisce un valore di intensità sogliato.

I *non flat* sono poco usati per difficoltà pratiche.

---

## EROSIONE FLAT E DILATAZIONE

**Erosione:** consiste nel considerare il valore di intensità minimo dell'immagine  $f$  quando considero come punti tutti quelli per cui aggiungo le posizioni all'interno dell'elemento strutturante.

**Dilatazione:** prevede anche la riflessione. Dobbiamo considerare che riflettiamo l'elemento strutturante, andiamo a ribaltare e facciamo la stessa operazione.

L'erosione e dilatazione flat su una scala di grigio, può essere usata per ridurre rumore ed eliminare parti non desiderate.

Con l'erosione scurisco l'immagine e le parti luminose si riducono, mentre le parti scure si ingrandiscono.

Con la dilatazione rendiamo le parti scure ridotte e le luminose ingrandite.

## EROSIONE E DILATAZIONE NON FLAT

**Erosione:** uguale ma al minimo sottraggo il valore del punto dello strutturante

**Dilatazione:** uguale ma al minimo aggiungo il valore del punto dello strutturante

## APERTURA E CHIUSURA FLAT

Uguale [vedi slide]

L'apertura diminuisce l'intensità di dettagli luminosi proporzionalmente alla dimensione dello strutturante.

Con la chiusura attenuo i dettagli scuri.

---

## SMOOTHING MORFOLOGICO

Non uso filtri spaziali o di frequenza ma sulla base della morfologia: se l'apertura sopprime dettagli luminosi più piccoli, l'apertura è come fare *smoothing* perché elimino rumore. La chiusura fa l'opposto, *sharpening*. Se combino i due operatori posso ridurre il rumore.

---

## GRADIENTE MORFOLOGICO

Il gradiente ci consente di evidenziare i contorni degli oggetti.

Qui possiamo usare dilatazione, erosione e sottrazione di immagine per ottenere il gradiente:

parte da *f* e ottengo il gradiente: *prendo l'immagine, la dilato e al risultato la sottraggo*.

La dilatazione ispessisce i dettagli dell'immagine, l'erosione li contrae, quando faccio la differenza enfatizzo i confini delle regioni, come fa il gradiente!

---

## TRASFORMAZIONI TOP-HAT & BOTTOM-HAT

Posso combinare apertura e chiusura con la sottrazione per ottenere trasformazioni top e bottom hat.

La top hat: sottrae all'immagine la sua apertura.

La bottom hat: prende la chiusura e le sottrae l'immagine.

Il senso di ciò è rimuovere oggetti usando elementi strutturanti tali per cui non si adattino all'oggetto da rimuovere.

**Il top-hat viene usata quando ho oggetti chiari su sfondo scuro, la bottom-hat al contrario.**

Un'applicazione importante la si ha per la *granulometria*.

Viene usata per correggere l'illuminazione: se ho un'immagine mal illuminata ho risultati falsi.

---

Ca' maronn t' accumpagn.