

PER ALTRI APPUNTI CONSULTARE IL SITO:
https://luigi-v.github.io/Appunti_Universita/

1. CAMPIONAMENTO E ISTOGRAMMA

Nel quadrato dove viene rappresentata una immagine del mondo reale, applicando una tecnica di **Image Processing**, si ricava una ulteriore immagine che differisce dalla precedente per una qualche motivazione, provando ad estrarre degli aspetti qualificanti (ad esempio radiografia in campo medico per evidenziare una anomalia), infatti la freccia presente in figura parte e arriva sulla stessa immagine.

Quando invece si effettua una operazione di **Computer Vision**, si collega un qualche aspetto dell'immagine del mondo reale, a qualcosa di più astratto quello che risiede nel mondo della conoscenza, automatizzando determinati passaggi.

La **Computer Vision** risulta essere una evoluzione rispetto all'**Image Processing**, dove quest'ultima non fa nient'altro che trasformare i pixel senza interpretarli. Invece la **Computer Vision** assegna a questi pixel un significato sulla base di una informazione fornita da un esperto, interpretando il contenuto semantico dell'immagine sulla base di un dominio di competenza.

La **modellazione geometrica** rappresenta oggetti astratti e la **Computer Graphics** li rende visibili sotto forma di immagine (si pensi alla progettazione di una nuova auto che passa dall'idea alla realizzazione grafica). Ad esempio, creare un t-rex con poligoni geometrici, visto che questi non esistono più e non è possibile campionarli, a differenza di un cane.

Da ciò si può capire che l'**Image Processing** risulta essere un'attività di **basso livello**, siccome essa non si addentra nella semantica dei pixel (non si entra nell'interpretazione di ciò che è presente nell'immagine) come le operazioni di *compressione, rappresentazione e trasmissione*.

Man mano che si va verso la **Computer Vision**, ovvero quelle tecniche di elaborazione che tentano di capire cosa è presente nell'immagine, considerata un'attività di **alto livello (interpretazione e predizione)**.

INTENSITÀ E LUMINOSITÀ:

- **Intensità dell'immagine:** è l'energia luminosa emessa da un singolo punto dell'immagine e dipende dal dispositivo. Ovvero un dispositivo in grado, in un singolo punto, di emettere una energia luminosa diversa, maggiore è l'intensità prodotta e più il device sarà sofisticato.
- **Luminosità dell'immagine:** il dispositivo emette una certa intensità luminosa che viene percepita dal soggetto sotto forma di luminosità. Essa è ciò che viene percepito della luminosità emessa da una singola unità (singolo pixel). Dipende dal contesto (è anche soggettiva), che la rendono variabile.

Un esempio relativo alla luminosità, ovvero alla capacità umana di percepire variazioni di luminosità è legata alla **Legge di Weber**. Esso posizionava un utente di fronte a uno schermo, il quale nella porta posteriore (non visibile all'utente) aveva un cannone luminoso che sparava una certa intensità luminosa. Succedeva che lo schermo veniva illuminato con una certa intensità I e il centro dello schermo veniva illuminato con $I + \delta(I)$. Inizialmente il centro dello schermo aveva la stessa intensità dello sfondo senza esser percepita nessuna differenza. Facendo variare le due intensità, si chiedeva all'utente fin quando fosse stato possibile individuare queste variazioni di intensità. Weber ha dedotto che l'essere umano possiede una buona capacità di discriminazione quando I è basso, quindi una luminosità scura. Man mano che I cresce la capacità nel notare variazioni di luminosità diventa meno forte. Quindi se si vuol percepire una variazione di luminosità con un I molto grande, deve essere prodotto un $\delta(I)$ molto grande. Mentre con un I molto piccolo si è in grado di percepire una variazione di luminosità anche molto piccola. Come si evince dal grafico, progressivamente che aumenta l'intensità dello sfondo, direttamente aumenta la variazione che dovrà essere prodotta per essere visibile.

Good Brightness Discrimination:
 $\Delta I / I = \text{small value}$

Poor Brightness Discrimination:
 $\Delta I / I = \text{large value}$

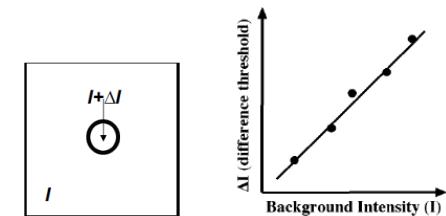
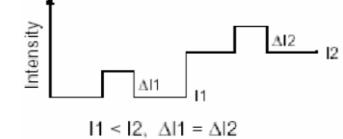


IMAGE CONTRAST:

Un altro concetto importante che spesso si confonde nel definire una immagine è il **contrasto di una immagine**.

Esso dovrebbe fornire una idea di quanto gli elementi presenti in una immagine sono distinti uno dall'altro.

Queste due immagini hanno un contrasto diverso da un punto di vista percettivo anche se dal punto di vista numerico no. L'immagine di sinistra presenta due rettangoli uno con uno sfondo uniforme ed un altro. All'interno vi è un altro rettangolo più chiaro. Seguendo l'intensità che rende l'idea di quale è la variazione tra lo sfondo e l'oggetto, lo sfondo si mantiene costante fin quando non si intercetta il rettangolo grigio, poi vi è un incremento ΔI che mantiene costante l'intensità per un certo periodo per poi decrescere e ritornare di nuovo sullo sfondo. Quindi la variazione indotta nel passare dallo sfondo all'oggetto è un ΔI , quindi una variazione di luminosità quantificabile in ΔI . Il secondo rettangolo è simile al primo ma con intensità diverse, partendo con una intensità più chiara, mantenendosi costante fin quando non viene intercettato il rettangolo più chiaro.



Quindi anche in questo caso vi è una variazione di luminosità ΔI costante per poi scendere nuovamente e mantenersi costante fino alla fine. La variazione indotta nel passaggio dallo sfondo all'oggetto è ΔI . Le due variazioni sono equivalenti $\Delta I_1 = \Delta I_2$, ma l'immagine di sinistra è meglio contrastata, in quanto l'oggetto centrale si nota in maniera più distinta rispetto allo sfondo, benché le variazioni siano equivalenti.

Quindi se si volesse misurare il **contrasto** di una immagine non si potrebbe semplicemente calcolare la variazione tra un oggetto e un altro (sfondo e oggetto centrale), ma va considerato un ulteriore elemento, ovvero una sorta di **normalizzazione**. Quindi si effettua la differenza tra l'intensità dell'oggetto e del contorno, e poi viene normalizzato tutto rispetto all'intensità del contorno.

$$C = \frac{|I_p - I_n|}{I_n}$$

$$C = \frac{0.3 - 0.1}{0.1} = 2 \quad C = \frac{0.7 - 0.5}{0.5} = 0.4$$

Per ottenere il valore numerico che evince quale immagine è meglio contrastata rispetto ad un'altra, si può normalizzare rispetto allo sfondo, come si vede dalla figura dove viene assegnato un valore alla luminosità per ogni rettangolo e relativo sfondo e poi vengono effettuate le operazioni di normalizzazione dividendo per il valore dello sfondo ottenendo due valori, risultati dalla formula in rosso, che permettono di individuare il contrasto di una immagine, in questo caso siccome $2 > 0.4$, si può dire che la prima immagine presenta un contrasto migliore all'immagine precedente.

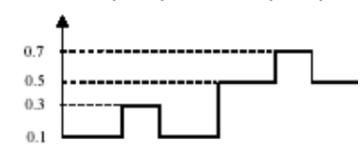
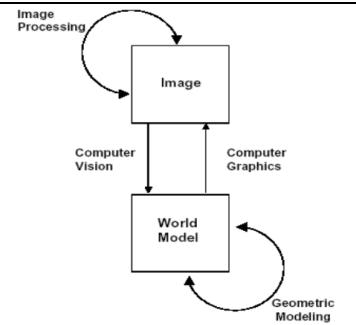


IMAGE FORMATION:

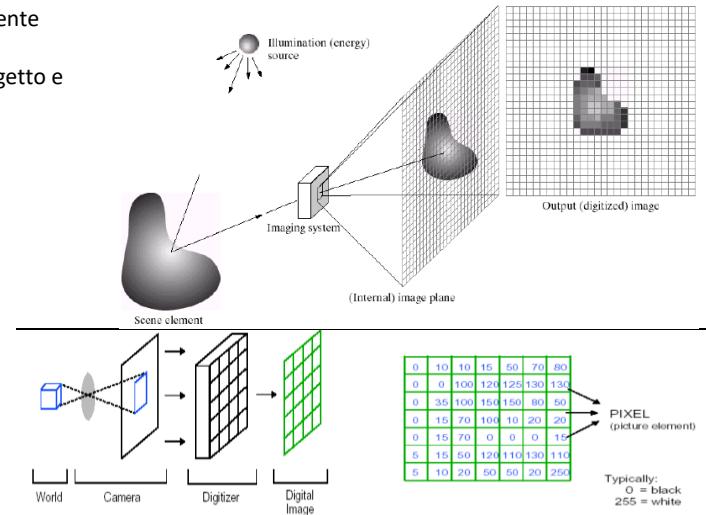
Il processo di **digitalizzazione di una immagine** risulta essere importante in quanto digitalizzando una immagine o un video, si perde inevitabilmente una parte dell'informazione. Quando si acquisisce una immagine con una singola sorgente (dispositivo) si passa da una rappresentazione tridimensionale ad una rappresentazione bidimensionale, e questo passaggio da 3D a 2D comporta irrimediabilmente una perdita di informazioni. Se tale processo non viene svolto nella maniera adeguata il processo di digitalizzazione viene aggravato da un ulteriore perdita di informazione.



Fondamentalmente, si può effettuare la digitalizzazione perché si ha una sorgente luminosa che illumina l'oggetto per effetto della riflessione dei raggi luminosi sull'oggetto, il sensore acquisisce le informazioni ottenute, di rimbalzo sull'oggetto e produce una rappresentazione digitalizzata dell'immagine.

$$f(x,y) = \text{reflectance}(x,y) * \text{illumination}(x,y)$$

Reflectance in [0,1], illumination in [0,inf]

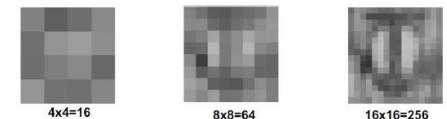


Come si vede dalla figura, vi è l'oggetto tridimensionale, che viene acquisito dal dispositivo il quale all'interno presenta una rete la quale, in ogni posizione prevede un sensore incaricato ad acquisire una porzione del mondo reale e trasformare tale acquisizione in termini numerici, calcolando la quantità di luce che viene riflessa in quella determinata posizione convertendola in numero, composta da pixel (acronimo di picture element).

CAMPIONAMENTO (SAMPLING):

L'operazione di **sampling** (campionamento progressivo) è l'operazione attraverso la quale si decide quanti sensori utilizzare nella fase di digitalizzazione. Quindi la fase di digitalizzazione è l'intero processo che viene diviso in due fasi, una di esse è il campionamento.

Come si evince dalla figura verranno utilizzati in modo progressivo sempre più **sensori** risultati della moltiplicazione tra numero di righe e numero di colonne. Man mano che aumentano i sensori e la porzione di acquisizione di ognuna di essa diventa sempre più piccola, l'immagine assume una qualità migliore.



QUANTIZZAZIONE:

L'altra fase del processo di digitalizzazione di una immagine è la quantizzazione. Mentre il campionamento concerne il **numero di sensori (pixel)** impegnati nella fase di acquisizione, la **quantizzazione** è legata al numero di livelli di **intensità luminosa (colori)** che ogni sensore può produrre.

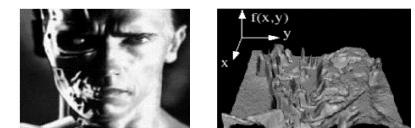
Nelle figure, viene assegnato per ogni sensore un determinato numero di bit. Ogni sensore converte la quantità di luce proveniente dal mondo reale in base al numero di bit assegnati. Man mano che aumentano i numeri di bit, di conseguenza aumentano il numero di livelli di intensità luminosa migliorando sensibilmente la qualità dell'immagine. Tutte le immagini hanno conservato un campionamento costante (non sono variati il numero di sensori) con quantizzazione variabile.



COSA È UNA IMMAGINE:

L'**immagine** è un insieme di informazioni luminose che vengono catturate da sensori, che possono essere convertite in informazioni tridimensionali.

Ad ogni posizione (x, y) nella rappresentazione di una immagine, corrisponde un valore $f(x, y)$ che risulta essere il livello di luminosità catturato dal sensore. La rappresentazione tridimensionale di una immagine in termini di $f(x, y)$ può essere quella dove i colori **più bassi** sono quelli tendenzialmente **più scuri**, man mano che si va verso una zona **più chiara** vi sono dei **rilievi**. Quindi il valore $f(x, y)$ tende a crescere dando l'effetto di tridimensionalità.



Si può immaginare l'immagine come un insieme di valori $f(x, y)$ dove ad ogni pixel (x, y) si assegna un valore di luminosità, che produce la rappresentazione tridimensionale.

Questa è la rappresentazione matriciale, presente in memoria, dove ogni valore corrisponde ad un livello di luminosità variabile a seconda se il sensore assegnato ad ogni pixel ha catturato un livello di intensità scuro o chiaro.

i	j	$f(x,y)$
62	79	23
10	10	9
10	58	197
176	135	5
2	1	1
0	89	144
255	252	0
166	63	127
		17
		1
		0
		99
		30

RISOLUZIONE:

La risoluzione lega il campionamento e la dimensione fisica (dell'oggetto, ad esempio a diverse distanze). Va divisa in due elementi:

- **Temporale** (per i video): definisce il numero di immagini al secondo da usare (FPS). Per la visione umana si usano da 15 a 30 immagini al secondo.
Al di sotto di tale valore vi è una bassa risoluzione temporale.
- **Spaziale** (singole immagini): definisce il numero di pixel da usare (in alternativa DPI, dot per inch: il numero di pixel al centimetro).

Queste due immagini hanno la stessa dimensione fisica ma sono profondamente diverse. A destra l'effetto di blocchettoizzazione è molto più evidente rispetto all'immagine di sinistra. Tale effetto è molto evidente dove vi sono i dettagli più importanti, infatti, l'immagine è stata acquisita con 72 DPI. L'immagine di sinistra acquisita con 150 DPI, risulta essere sensibilmente migliore e l'effetto di blocchettoizzazione è quasi totalmente scomparso. Quindi pur avendo due immagini di uguale dimensione, ma con un numero di pixel maggiore l'una rispetto all'altra, si evince come quest'ultima sia qualitativamente migliore, legata al numero di DPI.



IMAGE PROCESSING:

Una immagine è assimilabile a una funzione tridimensionale $f(x, y)$, quando si effettua una operazione di **Image Processing** che definisce una nuova immagine g in termini di una immagine esistente f , effettuando una **trasformazione** t . Si trasforma un pixel, non interpretando il contenuto dell'immagine, ma viene effettuata una trasformazione t su tale pixel per trasformarlo in un altro pixel.

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

Si possono effettuare varie tipologie di trasformazioni sull'immagine, sulla base anche delle immagini che si hanno a disposizione, come:

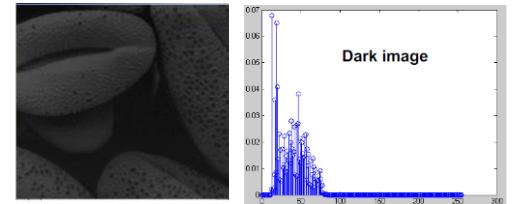
- **Immagini Binarie**, ovvero in bianco e nero, con un bit per pixel.
- **Immagini in Scala di Grigio**, oltre al bianco e il nero vi sono varie sfumature di grigio, e la quantità di sfumature in termini di livelli di grigio, dipendono dalla quantizzazione. Una quantizzazione a 8 bit, si avranno 256 colori, dove 0 è nero, 255 bianco e le restanti sono sfumature di grigio.
- **Immagini a colori**, dove si compone ogni singolo pixel attraverso una serie di componenti che dipendono dal modello di colore che si utilizza.

IMAGE ENHANCEMENT (MIGLIORAMENTO DELL'IMMAGINE) - HISTOGRAM BASED METHODS:

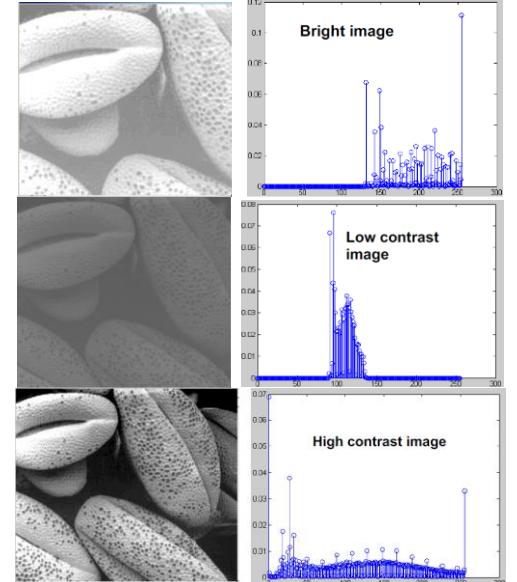
Una misura importante che può essere assegnata all'immagine è quella dell'**istogramma**, che è una rappresentazione statistica dell'immagine, attraverso il quale è possibile avere una distribuzione di quelli che sono i livelli di grigio. Si ha il numero di occorrenze per ogni tipologia di colore presente nell'immagine.

Sull'asse delle ascisse viene rappresentato il **risultato della quantizzazione**, ovvero il numero di colori rappresentabili. Poiché questa immagine è stata digitalizzata usando 8 bit per pixel, sull'asse delle ascisse ci saranno 256 colori.

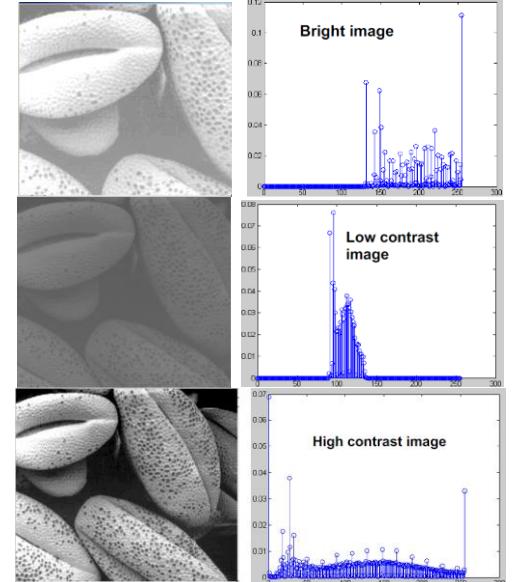
L'asse delle ordinate riporta il **numero di occorrenze** per ognuno dei colori. Ciò significa che ognuno dei picchi rappresenta quante volte il colore corrispondente è rappresentato nell'immagine. Dato che l'immagine presenta una prevalenza di colori scuri, di conseguenza nell'istogramma vi sarà una concentrazione di picchi nell'intervallo di colori che varia da 0 a 75, e poche occorrenze di colori chiari (fortemente addensato a sinistra).



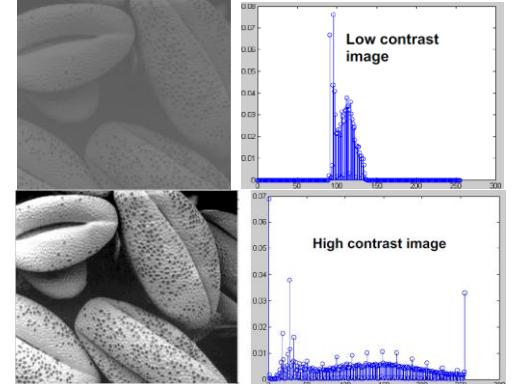
Viceversa, in questa immagine dove si ha una prevalenza di colori chiari, molto luminosa, vi sono dei picchi nella parte destra, ovvero nella parte chiara.



In precedenza, quando si è parlato del contrasto, si è detto che in presenza di una immagine poco contrastata è possibile intervenire per poterla trasformare. Per capire tale concetto, ovvero che una foto è **poco contrastata**, l'istogramma assumerà la forma in figura, con una campana molto stretta concentrata in pochi colori.



Quando un istogramma si presenta in questa modalità, si può effettuare una operazione di **equalizzazione** dell'istogramma, andando a *stiracchiare* l'istogramma per ottenere una redistribuzione dei colori, avendo in output una immagine ben contrastata.



Per effettuare una **equalizzazione**, si è detto che tale operazione trasforma l'istogramma iniziale G in un altro istogramma G (cappelletto):

$$\hat{G} = T(G)$$

Una volta acquisita l'immagine e aver preso coscienza della campana relativamente stretta, si applica la trasformazione T all'istogramma per produrre il nuovo istogramma G (cappelletto). Questa trasformazione T deve seguire tre regole tassonomiche (*ordine di importanza*):

1. L'istogramma deve essere il **più piatto possibile**, senza grossa variazione tra le occorrenze.
2. **L'ordine dei livelli di grigi** deve essere mantenuto, con piccole oscillazioni.
3. L'istogramma deve essere **non frammentato**, non vi devono essere dei "salti" (quando si salta un colore).

EQUALIZZAZIONE DELL'ISTOGRAMMA:

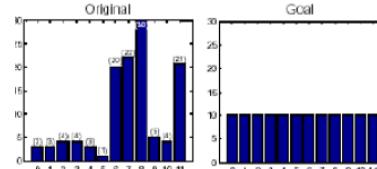
L'equalizzazione non è nient'altro che una rappresentazione statistica, per conoscere la ricorrenza in termini di rappresentazione di ogni colore. Poiché ogni colore rappresenta un certo numero di pixel, la somma delle occorrenze di tutti i colori darà come risultato il numero totale di pixel. Si tratta di rendere l'istogramma il più piatto possibile. Questo potrebbe rendere l'immagine più leggibile, anche se la cosa è opinabile (in alcuni casi). Supponiamo che l'immagine di partenza sia di dimensioni 512x512, ovvero che sia composta da 2^{18} punti (numero dei campioni). Questo significa che, detto $H(i)$ l'istogramma (somma dei valori sull'asse delle ordinate), deve valere che:

$$\sum_{i=0}^{255} H(i) = 2^{18}$$

Se si vuol rendere l'istogramma piatto significa che si vuole:

$$H(i) = 2^{18}/256 = 2^{10} = 1024 \forall i,$$

dove 2^{18} sono tutte le occorrenze e 256 è la quantizzazione fatta a 8 bit (asse x = 256 colori), se si vuole ottenere una equalizzazione ideale, per ogni colore si dovrà avere 2^{10} come frequenza, significa che ogni colore dovrebbe essere più o meno presente 1024 volte.

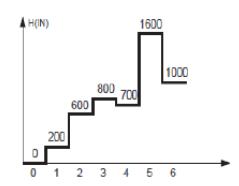


Naturalmente non si potrà ottenere un'esatta uguaglianza, visto i limiti dell'elaborazione, ma si tenta di avvicinarsi il più possibile. Nel primo livello di uscita si vorrebbe ottenere 1024: se si sommano i primi tre livelli (0, 1 e 2) dell'ingresso otteniamo 800, se si sommano i primi quattro 1600. Poiché 800 è più vicino a 1024 di 1600 scegliamo di sommare i primi tre.

Passando al secondo livello dell'uscita: anche lui dovrebbe essere riempito con 1024 livelli ma, per evitare accumuli dell'errore, bisogna considerare che nei primi due livelli voglio $1024+1024=2048$.

Poiché nel primo ne sono già stati inseriti 800 si dovrà cercare di approssimare, per il secondo, il valore di 1248. Quindi si sceglie di inserire i livelli 3 e 4 dell'ingresso, visto che $800 + 700 = 1500$ è più vicino a 1248 di 800. Nel terzo livello di uscita ci va ancora 1024 e la somma dei primi tre deve essere 3072. Ne sono stati già sistemati $800 + 1500 = 2300$ e quindi ne restano 772. Poiché il primo livello da sistemare dell'ingresso (il 5) vale 1600 e $1600-772 = 828$ si lascia vuoto il terzo livello dell'uscita. Nel quarto livello di uscita devo approssimare $4096 - 2300 = 1996$, se viene inserito il livello di ingresso 5 ne mancano 396, si inserisce 5+6 ne avanzano $2600 - 1996 = 604$ quindi scelgo 5.

IN	OUT
0	0
1	0
2	0
3	1
4	1
5	3
...	...



2. IMAGE FILTERING

In generale una trasformazione dell'immagine è caratterizzata nel seguente modo:

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

dove f è l'immagine input, g è l'immagine elaborata, T è l'operatore di trasformazione applicata ai singoli pixel dell'immagine.

L'operatore che si applica all'immagine può avere diverse caratteristiche matematiche. In genere si possono definire degli operatori lineari se godono di determinate proprietà come:

- **Omogeneo** se $T\{af[n]\} = aT\{f[n]\}$
- **Associativo** se $T\{f_1[n] + f_2[n]\} = T\{f_1[n]\} + T\{f_2[n]\}$

Il fatto che un operatore possa essere lineare o meno non ha un impatto sulla tipologia di trasformazione che si va ad applicare.

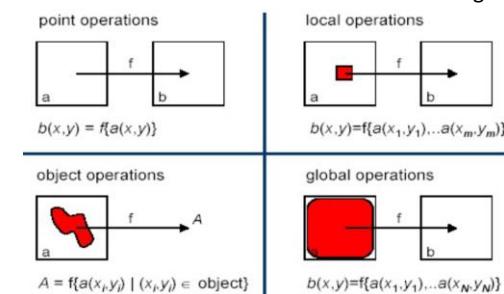
CLASSIFICAZIONE DELLE TECNICHE (O TRASFORMAZIONI):

Innanzitutto, va fatta una classificazione delle trasformazioni T che possono essere applicate ad una determinata immagine/video:

- Tecniche operanti nel **dominio spaziale**, ovvero dove vi è una manipolazione diretta dei pixel.
- Tecniche operanti nel **dominio delle frequenze** (trasformata di Fourier), ovvero vi è una trasformazione dell'immagine nel dominio delle frequenze e conseguente manipolazione di esse. Tali frequenze non agiscono sui pixel ma su una trasformazione dei pixel su un concetto diverso. Tali tecniche implicano un costo computazionale aggiuntivo compensato da una qualità della trasformazione dell'immagine migliore.
- Tecniche **ibride**, ovvero una combinazione di varie tecniche.

L'altro elemento che si può aggiungere nella classificazione delle tecniche è quello che permette di dividere le tecniche di trasformazione in 4 famiglie:

- **Point Operations**: le tecniche che rientrano in questa classe risultano essere molto semplici, ovvero trasformazioni dove il pixel di destinazione, nella trasformazione, dipende solo dal pixel di partenza.
- **Local Operations**: hanno una struttura più aggregata, dove il pixel di destinazione non dipende solo da un pixel di partenza, ma anche da un suo intorno quadrato.
- **Object Operations**: il pixel di destinazione dipende dal pixel di partenza e da un oggetto che ha un suo significato semantico, non di forma regolare come nel caso delle *Local Operations*.
- **Global Operations**: ogni pixel dell'immagine di destinazione, dipende da tutti i pixel dell'immagine di partenza.



TECNICHE OPERANTI NEL DOMINIO SPAZIALE:

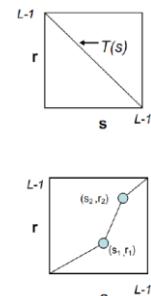
Partendo con il **Point Processing**, dove ogni pixel di destinazione dipende solo dal corrispettivo pixel di partenza.

In questa rappresentazione grafica, sull'asse delle ascisse si hanno gli input gray level (livelli di grigio dell'immagine di partenza), r , invece rappresenta il valore output e T effettua la trasformazione. Per ogni pixel situato su s , T ne ottiene una trasformazione che lo porta ad assumere un nuovo valore lungo la retta $T(s)$, ad esempio il colore 0 in s si trasformerà nel colore 255 in r tramite $T(s)$.

Questa trasformazione che si sta effettuando, $T(s)$, è una trasformazione monotona decrescente, che prende il nome di **Negativo**.

È possibile apporre anche altre operazioni come la **Binarizzazione** (T è monotona crescente) di una immagine. In essa i valori ottenuti in fase di digitalizzazione, mappati sull'asse delle ascisse, sono valori che devono essere distribuiti soltanto su due colori, ovvero il bianco e il nero. Ottenendo un'immagine di output, appunto, bianco e nera.

Oltre alla binarizzazione, agendo sui due cerchi (a destra) è possibile anche effettuare il **Contrast Stretching** dell'immagine, operando direttamente sull'istogramma per migliorare il contrasto dell'immagine in questione.



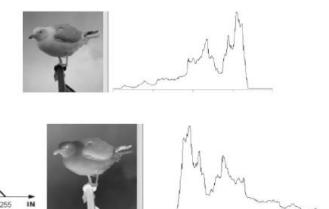
NEGATIVO (TRASFORMAZIONE POINT OPERATORS):

L'istogramma, si ricordi, è rappresentato da un'asse delle ascisse dove vi sono tutti i livelli di grigio frutto della quantizzazione, e sull'asse delle ordinate vi è il numero di occorrenze per ognuno dei livelli di grigio. Produrre il **negativo** di tale immagine, significa associare ad ogni pixel il livello di grigio complementare.

Infatti, tra le due immagini intercorre una differenza specifica: i pixel chiari sono diventati più scuri e viceversa. La trasformazione quindi in presenza di colori scuri si occupa di tramutarli in colori chiari.

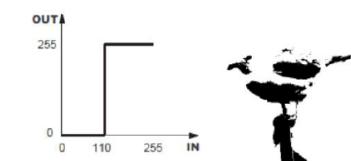
Progressivamente, sull'asse delle ascisse, i colori diventando più chiari, e con la progressione della funzione T , essa tende a renderli più scuri ottenendo il negativo dell'immagine.

L'istogramma risultante è ad immagine speculare di quello di partenza (ribaltato rispetto al valore mediano).



BINARIZZAZIONE (TRASFORMAZIONE POINT OPERATORS):

Questa trasformazione individua una soglia in 110 (variabile), dove tutti i colori da 0 fino a 110 diventeranno nero, e tutti i colori che partono da 110 fino a 255 diventeranno bianco. Ottenendo la stessa immagine di partenza, ma **binarizzata**.



FILTRI SPAZIALI:

Con i **filtri spaziali (mask processing)** è possibile effettuare varie operazioni. Esistono tre categorie di filtri che si andranno ad applicare alle immagini:

- **Lowpass** (passa basso): attenua o elimina le alte frequenze, ovvero **contorni e dettagli**.
- **Highpass** (passa alto): attenua o elimina le basse frequenze, come **contrasto e intensità**.
- **Bandpass** (passa banda): attenua frequenze in una banda predefinita.

Questi filtri rientrano nella categoria dei **Local Operations** dove ogni pixel di destinazione dipende dal pixel di partenza più il suo intorno, ovvero la trasformazione T è definita in un intorno predefinito di (x, y) .

Per essere costruiti, in genere vengono create delle mascherine quadrate, generalmente di dimensioni dispari 3x3, 5x5 ecc... e dentro tali mascherine verranno posti dei pesi W_i , che chiaramente andranno a determinare la funzionalità del filtro. Sulla base dei valori $W_1 \dots W_9$, distribuiti all'interno della mascherina si otterrà un risultato diverso. Una volta costruita, essa si sovrapporrà all'immagine centrando nel pixel (x, y) che avrà come corrispondente il peso W_5 e si effettuerà la moltiplicazione tra ciascuno di questi pesi con il corrispondente valore di luminosità presente nell'immagine:

$$R = w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + w_7z_7 + w_8z_8 + w_9z_9$$

dove z_1, \dots, z_9 rappresentano i pixel dell'immagine ai quali è sovrapposto il filtro e il valore R sarà quello proiettato nell'immagine di destinazione nella posizione (x, y) .

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Nell'immagine viene presa una zona specifica che rappresenta vari pixel. Si applica una maschera composta da tutti 1 ed un fattore di normalizzazione di 1/9. La maschera con tutti 1 sarà sovrapposta a quella dell'immagine iniziale e dopo i vari calcoli, si otterrà la media della zona in questione. Il pixel in questione sarebbe la media della porzione dell'immagine considerata.

Questa prima classe di filtri verrà utilizzata per costruire i cosiddetti ***Smoothing Filter***, cioè i filtri utilizzati nelle operazioni di preprocessing. Infatti, può capitare che l'immagine contenga del "rumore" oppure sia troppo ricca di dettagli. Con tali filtri è possibile intervenire, anche se essi possono produrre un effetto di ***Blurring*** se l'immagine è troppo ricca di dettagli, e quindi se ne vogliono rimuovere alcuni di essi per rendere l'immagine più funzionale per l'estrazione di oggetti.

Oppure l'immagine presenta del rumore che può esser stato introdotto nella fase di digitalizzazione o durante la fase di trasmissione. Questi filtri ricadono nella categoria dei filtri ***Lowpass*** (bassa passo) ovvero i filtri che lasciano transitare solo le informazioni che risiedono nelle basse frequenze come contrasto e intensità, i filtri Lowpass si dividono in ***lineari*** (Filtro Media e Laplaciani) e ***non lineari*** (Filtro Mediano, Max, Min e Sobel).

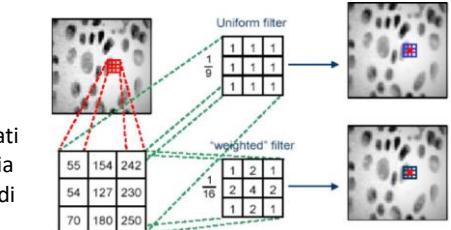
FILTRO SPAZIALE - MEDIA:

Tale filtro non differisce dal concetto di media aritmetica. Quindi ogni pixel dell'immagine destinazione viene sostituito con la ***media dei pixel*** appartenenti ad un'area ben definita dell'immagine. Piuttosto che prendere soltanto il pixel (x, y) nell'immagine input, viene preso la media di tutti i pixel connessi con il pixel (x, y) . Effettuando la media (con la divisione per 9) ottenendo il filtro di destinazione.

L'applicazione di un filtro media ad un'immagine/video che presenta rumore viene effettuato proprio per eliminare tale rumore prodotto da un determinato motivo, per ricondurre l'immagine ad una qualità migliore. Applicando un filtro media 3×3 si inizia col notare che parte del rumore e dei dettagli sono scomparsi, sfumando l'immagine. Utilizzando un filtro più di dimensioni maggiori, quando si esegue la media di un intorno più grande, in output vi sarà una eliminazione del rumore e dei dettagli. Il progressivo aumento della dimensione del filtro, fa sì che il rumore venisse diminuito, ma ha comportato anche un effetto di eliminazione delle informazioni più significative presenti nell'immagine come contorni e dettagli.

Più cresce il filtro più aumenta l'impatto delle informazioni eliminate.

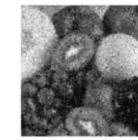
Il ***contorno*** di un oggetto è una ***transizione***, tra un oggetto e un altro, come una sorta di interruzione.



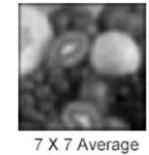
- $S =$ insieme di pixel connessi con (x, y)
- $M =$ numero di pixel coinvolti ($|S|+1$)

$$g(x, y) = \frac{1}{M} \sum_{f(n,m) \in S} f(n, m)$$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Salt & Pepper Noise



7 X 7 Average

FILTRO SPAZIALE – MEDIA ADATTIVA:

È possibile modificare il filtro media per ottenere un risultato qualitativamente migliore. La ***media adattiva*** va a risolvere varie problematiche della media, come la confusione dei contorni. Ogni qual volta essa viene utilizzata effettua un confronto tra il pixel (x, y) e tutti quanti i pixel che sono nel suo intorno. In presenza di piccole variazioni si applica la media, ma in presenza di grandi variazioni (in base ad una ***soglia*** T) si lasciano invariati i pixel di riferimento. Essa ***lavora sui bordi***.

Applicando un filtro media sempre più grande, l'immagine tende a sfocarsi e perdere rumore, ma anche dettagli. Quindi la media adattiva può essere una soluzione che consente di preservare in parte dettagli e i contorni degli oggetti avendo un buon effetto sulla rimozione del rumore.

FILTRO SPAZIALE (NON LINEARE) – MEDIANA:

Un'altra possibile soluzione che può essere utilizzata per rimuovere il rumore è il ***Filtro Mediana***. Esso corrisponde esattamente al concetto di mediana statistica per il quale si considerano i pixel di una regione, li si ordina in maniera crescente e poi ci si posiziona sul valore centrale.

$$g(x, y) = \text{mediana}\{f(x, y)\}_{f(x, y) \in S}$$

La mediana è proprio tale valore centrale. Rispetto alla media, la mediana ***non viene influenzata dalle code***, ovvero il valore centrale non viene condizionato particolarmente dalla presenza di significativi cambiamenti agli estremi, cosa che avviene nella media. Quando viene applicata la mediana ad un'immagine con rumore, si ottengono i seguenti risultati.

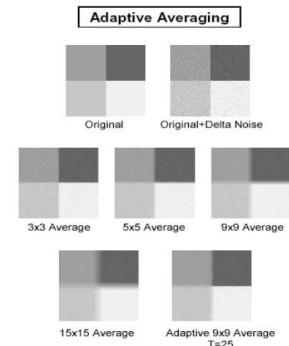
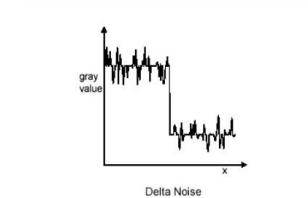
Invece che aumentare la dimensione del filtro mediana, si reitera lo stesso filtro. Si parte sempre da filtri 3×3 . Quindi si prende tale filtro in cui si è calcolata la mediana e lo si è applicato all'immagine.

Successivamente lo si applica più volte, così facendo si determina un filtraggio progressivo ma utilizzando sempre lo stesso filtro, mantenendo la stessa dimensione. Si può notare che all'immagine, pur essendoci alterazione sui bordi, l'effetto ***Blurring*** è molto meno evidente rispetto alla media tradizionale.

Alle volte è possibile combinare media e mediana, ottenendo dei ***filtri ibridi***.

$$S = \text{neighborhood of pixel } (x, y) \\ M = \text{number of pixels in neighborhood } S \\ T = \text{predefined threshold value}$$

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(n,m) \in S} f(n, m) & \text{if } \frac{1}{M} \sum_{(n,m) \in S} |f(n, m) - f(x, y)| < T \\ f(x, y) & \text{otherwise} \end{cases}$$



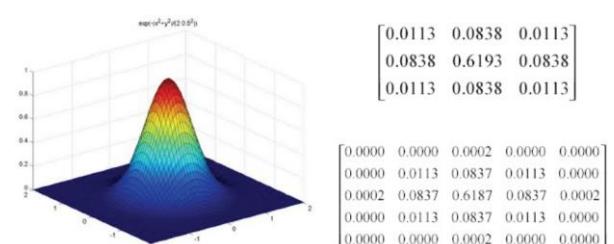
MEDIA VS MEDIANA:

La ***Media*** presenta una ***eccessiva velatura (Blurring)*** al crescere della maschera e delle iterazioni con conseguente perdita dei dettagli, essa però ***preserva***, con la versione adattive, le forme (scalini). La ***Mediana*** ha una ***riduzione del rumore (noise reduction)*** senza eccessiva perdita di dettagli (***Blurring limitato***), ma non preserva le forme (scalini).

SMOOTHING GAUSSIANO:

Il funzionamento del ***Filtro Gaussiano*** è simile a quello del filtro mediana con la differenza che il contributo di ciascun ***pixel limitrofo ha un peso diverso***, definito dalla distribuzione spaziale della gaussiana. Nel campo dell'elaborazione delle immagini la gaussiana viene troncata limitandone l'estensione ad una zona di dimensioni $N \times N$ (i valori più frequenti per N sono 3, 4 e 5).

Si parte dal presupposto che i pixel in prossimità della zona centrale, sono quelli più importanti. Quindi se si vuol posizionare la maschera su un pixel (x, y) , non si deve assegnare poi lo stesso peso a tutti gli altri pixel ma si cerca di concentrare la maggior parte del peso in prossimità del pixel (x, y) , e progressivamente diminuire il peso dei pixel man mano che ci si allontana dal centro. La somma di tutti i pesi sarà ***sempre uguale a uno***. In prossimità del pixel di riferimento, si assegnerà un valore maggiore, in quanto c'è più probabilità che essi appartengono al pixel kernel.



Quindi possono essere costruiti nuovi filtri che sono dei **Filtri Media pesati**. Mentre prima il peso assegnato dalla media era uguale per tutti quanti, ora si adopera una selezione, più ci si è vicini alla posizione (x, y) maggiore sarà il peso in quanto maggiore sarà la probabilità che quel pixel appartenga alla regione di interesse, più ci si allontana dal pixel (x, y) maggiore sarà la probabilità di non appartenere alla regione di interesse e quindi si cerca di contaminare il meno possibile la trasformazione. Con un Filtro gaussiano si riesce ad eliminare le zone di rumore e a preservare gli scalini, proprio perché in prossimità di essi i pixel otterranno una distribuzione di pesi differenziata.

SUSAN - SMOOTHING SEGMENT UNIVALEUR ASSIMILATING VALUES:

Tale filtro esaspera il concetto Gaussiano. Infatti, permane l'idea di **Filtro Gaussiano** dove però i pesi non vengono stabiliti a priori, come fatto in precedenza, ma in questo caso i pesi vengono calcolati **run-time**. Quindi viene applicata la maschera all'immagine e si va a controllare quali pixel ricadono nella stessa area del pixel centrale. Si assegneranno pesi alti a tutti i pixel che appartengono all'aerea del pixel centrale e si tenderà a dare peso minore ai pixel che si trovano più in lontananza rispetto al centro. Questo calcolo lo si effettua per mezzo della formula seguente:

$$c(r, r_0) = e^{-\left(\frac{|I(r) - I(r_0)|^2}{t}\right)}$$

in cui $I(r)$ è il valore di intensità dei pixel all'interno della maschera, $I(r_0)$ quello del nucleo, t è un valore di soglia sull'intensità (fattore di normalizzazione) e $c(r, r_0)$ rappresenta il peso che il pixel in esame ha nel calcolo della media. Più i pixel sono diversi, più il peso va verso 0.

Oltre al peso dovuto alla funzione $c(r, r_0)$, per effettuare il filtraggio del rumore, si utilizza una funzione di tipo gaussiano. Se l'area dell'USAN è **nulla** il valore dell'intensità del pixel in esame è calcolato applicando il filtro mediano descritto precedentemente. Quindi si avrà ancora un filtro gaussiano, ma sarà una maschera in cui i pesi saranno calcolati di volta in volta sulla base dell'immagine. Il **vantaggio** di questo filtro è che i pesi dipenderanno dall'immagine e quindi non sarà un filtro statico costruito in partenza. Il **contro** di questo filtro è l'onerosità in termini computazionali.

Nell'immagine seguente viene vista una rappresentazione tridimensionale dei vari filtri applicati a delle immagini. Il filtro SUSAN **preserva gli scalini**, risultando essere un filtro ottimale fornendo output migliori.

Anche il filtro mediano garantisce tali aspetti, soprattutto in presenza di scalini. Uno dei problemi fondamentali è quella di preservare la distanza tra due oggetti, cosa che il filtro mediano riesce ad ottenere a differenza del filtro media. Quest'ultima non presenta un distacco tra un oggetto ed un altro, causando l'effetto di velatura molto evidente all'interno delle immagini e che induce una perdita dei contrasti.

EDGE DETECTION:

Si introduce un'altra classe di filtri che hanno come obiettivo quello di trattenere le informazioni che risiedono nelle alte frequenze, lavorando in modo esattamente speculare a quelli precedenti. Tali filtri trattengono i punti di controllo, ovvero i **dettagli e contorni**.

Gli **Edge Detector** sono filtri che hanno la capacità di individuare le aree di transizione tra regioni omogenee:

- **Punti isolati**
- **Linee** (orizzontali, verticali, oblique)
- **Contorni** (insieme di punti con inizio e fine coincidenti)

Gli edge possono avere forme differenti, tendenzialmente si identificano come una **variazione** di una serie di pixel o anche un singolo pixel, che ha una caratteristica ben definita, ovvero si passa da un'area a **transizione costante** (vicina alla 0) ad un'area in cui c'è una **variazione** (o perturbazione). Questa variazione può avere varie forme: rampa, scalino, ecc...

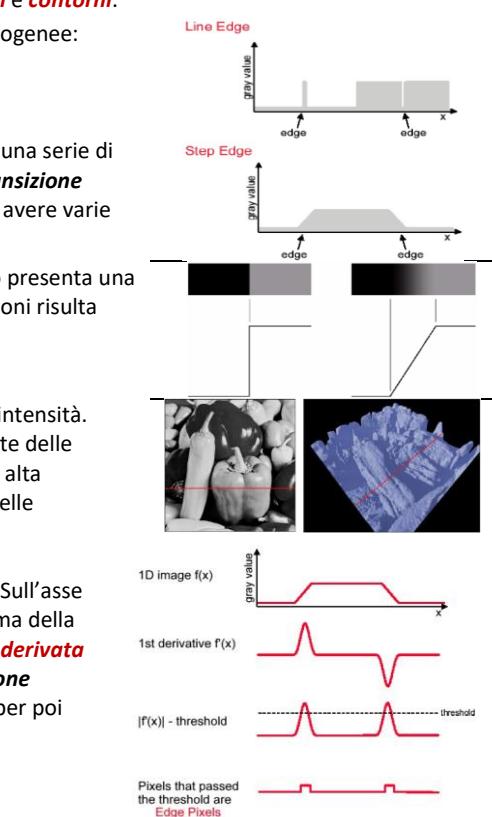
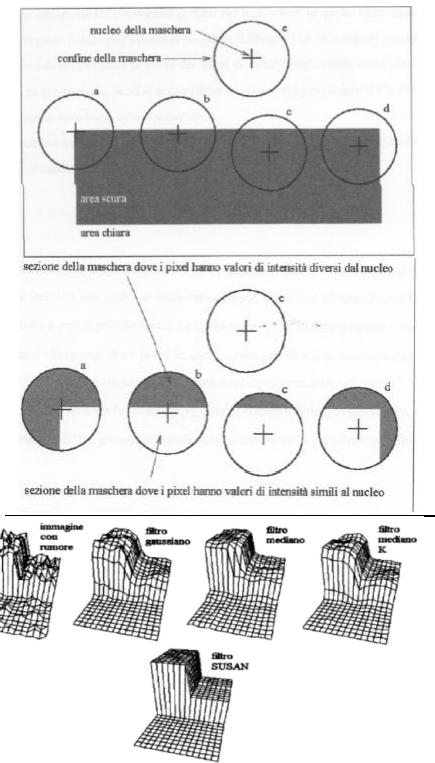
Tra le possibili situazioni di transizione tipiche, vi sono quelle a **rampa** e **scalino**. Il contorno a scalino presenta una transizione brusca tra due zone uniformi, invece nella transizione a rampa il passaggio tra le due regioni risulta essere più attenuato.

Nella figura vi è una immagine e la propria rappresentazione tridimensionale sotto forma di livelli di intensità. L'obiettivo è quello di individuare una sorta di detector che vada a mettere in evidenza le cime più alte delle variazioni di luminosità. La linea in rosso ha il compito di individuare quelle transizioni che hanno più alta probabilità di essere dei punti di contorno, non delle semplici variazioni all'interno dell'oggetto ma delle **variazioni significative** che indicano una transizione da un oggetto ad un altro.

Si supponga di avere una rappresentazione di un'immagine con una certa variabilità di toni di grigio. Sull'asse dell'ascisse vi sono i pixel e sull'asse delle ordinate i valori di grigio. Se viene calcolata la derivata prima della funzione in questione, si avrà come risultato delle informazioni sulla presenza di variazioni, quindi la **derivata prima** risulta essere **uguale a zero** quando **non vi sono valori significativi** e risulta avere una **variazione** laddove **esiste una variazione dei livelli**. Ritorna ad essere costante in presenza della zona uniforme, per poi trovare un'altra variazione e terminare in modo uniforme.

Quello che si vuol fare è capire se questi punti di massimo e minimo individuati dalla derivata prima, possono essere significativi. Quindi si vuol individuare una soglia che può dare informazioni circa i punti individuati al fine di comprendere se possono essere variazioni utili ad individuare un passaggio da un'area uniforme ad un'altra. Questi pixel presi singolarmente o in modo aggregato, sono quelli che possono essere potenzialmente etichettati come pixel di transizione, ovvero **Edge Pixels**.

La difficoltà fondamentale è capire la soglia utile, e se essa può essere resa universale oppure deve essere adattata a classi di immagini, e se viene individuata per classi di immagini si può esser sicuri che questa soglia vada bene per tutta quanta l'immagine. Ovviamente **NO** in quanto la soglia è molto difficile da individuare e non sempre quella individuata per classi di immagini vada bene. Spesso tale soglia all'interno della stessa immagine permette sia di individuare pixel significativi ma individua potenziali pixel di contorno che non risultano esserlo. Questo pone un limite dell'applicazione di tali filtri alle immagini digitali.



EDGE DETECTION – PUNTI ISOLATI:

I **punti isolati** sono le discontinuità più semplici da individuare, in quanto non contribuiscono a definire la **morfologia** di un oggetto. Il punto corrispondente alla posizione centrale della maschera può essere considerato un punto isolato quando, per esempio con riferimento alla maschera mostrata si ha: $R > T$, essendo T un opportuno valore di soglia (non negativo). Anche in questo caso, quando si individua un punto isolato affinché tale punto possa essere etichettato come punto di discontinuità, deve esserci una soglia T che viene individuata. Tale soglia T deve far sì che una volta applicata la maschera ad una determinata area ed una volta calcolato il valore R , si è in grado di etichettare il pixel a cui corrisponde R come pixel di edge, soltanto se il valore è abbastanza significativo. Quindi il valore di grigio di quel punto isolato deve essere **abbastanza differente** da quello dei suoi vicini. Se non vi è tale differenza evidente, non si può etichettare questo come punto isolato.

In figura un esempio, in cui vi è una prima possibile rappresentazione di un filtro di **Edge Detector** basato sulla derivata prima. La maschera 3x3 presenta **valori negativi** e un **valore positivo centrale**, e la somma di tutti valori corrisponde a 0. Quindi in linea col concetto di derivata prima, questo filtro deve garantire che quando viene applicato all'immagine si dovrà avere che nelle zone uniformi, il risultato deve essere zero, quindi il pixel kernel non è un pixel candidato ad essere un pixel di contorno, quando invece si trovano delle variazioni, esse devono essere evidenziate.

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 100 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} * \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & -95 & -95 & -95 \\ 0 & 0 & -95 & 760 & -95 \\ 0 & 0 & -95 & -95 & -95 \end{bmatrix}$$

Il principio di tali maschere è che la **somma dei pesi deve essere uguale a zero**, per assicurare il concetto di rapporto incrementale. Quest'ultimo sarà nullo nelle aree in cui non c'è variazione, e ci sarà un rapporto incrementale evidente positivo o negativo quando esiste una variazione.

EDGE DETECTION – LINEE:

È possibile costruire dei filtri che sono capaci di individuare non soltanto i punti, ma anche le **linee**, quindi privilegiare quei pixel che sono presenti lungo linee uniformi di variazione. Come il filtro precedente, hanno la caratteristica di avere pesi negativi e positivi, e la somma di tutti i pesi è uguale a zero. Ognuno dei filtri riesce ad individuare una particolare caratteristica di linea.

La prima maschera risponde più intensamente alle linee orizzontali (spesse un pixel): con uno sfondo costante, il massimo di R si ha infatti quando la linea coincide con la riga centrale della maschera.

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Analogamente, la seconda maschera risponde meglio a linee orientate a 45°, la terza a linee verticali e la quarta a linee orientate a 45°.



Si supponga di passare tutte e quattro le maschere su una immagine, e che in un certo punto si avranno R_1, R_2, R_3, R_4 risposte diverse delle quattro maschere. Per capire se il pixel appartiene potenzialmente ad una maschera bisogna verificare quale valore R_i risulta essere più grande rispetto agli altri, una volta individuato tale valore, allora il punto di edge che si sta etichettando è associato alla maschera R_i e si trova lungo una linea specifica di quella determinata maschera.

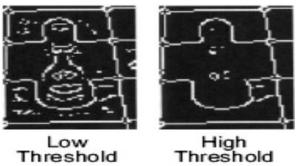
EDGE DETECTION – CONTORNO:

Il **contorno** (edge) di un oggetto rappresenta la **separazione** tra l'oggetto e lo sfondo o tra l'oggetto ed altri oggetti, per cui la sua estrazione è molto spesso il primo passo verso l'individuazione dell'oggetto. Un **edge** si presenta in una immagine come il confine tra due regioni caratterizzate da proprietà dei livelli di grigio in qualche modo distinguibili. Nel seguito si ipotizzerà che le regioni in questione siano sufficientemente omogenee, di modo che la determinazione della transizione tra le due regioni sia possibile sulla sola base della discontinuità dei valori di grigio. Le prime tecniche di **edge detection** che verranno analizzate sono basate sull'applicazione di un operatore locale di derivata. Per rendere le regioni sufficientemente omogenee tra loro è possibile applicare uno **Smoothing** in modo che vengano individuate soltanto le variazioni importanti. Per costruire dei filtri di edge detector basati sul principio di derivata è molto semplice, si prende una mascherina di dimensione dispari 3x3, 5x5, ecc... si assegnano i pesi in modo tale che essi siano tra di loro bilanciati con valori positivi e negativi e assicurando che la somma sia uguale a zero.

Uno dei problemi che si presenta è quello di stabilire se è necessario introdurre una **soglia**, che è una delle problematiche più importanti nelle fasi di costruzione degli edge detector. Nella figura vi è una immagine qualitativamente scadente, ed effettuare un'operazione di edge detector su questa immagine è piuttosto difficile estraendo il profilo degli oggetti che sono presenti. La forma del pedone bianco è facilmente ottenibile siccome essa riesce a distinguersi all'interno della scacchiera. Il pedone nero invece, si confonde con il tassello nero alle sue spalle. Quindi la determina della soglia che permette di tirar fuori il profilo dell'oggetto è abbastanza complicata. I progettisti di edge detector hanno a disposizione questa soglia e possono decidere se utilizzare una soglia alta o bassa e questo produce degli effetti differenti nella qualità dell'immagine. Un edge detector trasforma un'immagine in una **immagine bianco e nera** quindi con solo due livelli di grigio, in quanto un edge detector deve solo stabilire se ogni pixel di partenza è un pixel di contorno oppure no. Se un pixel **non è di contorno** viene etichettato come **nero** (valore 0), se un pixel **è di contorno** viene etichettato come **bianco** (valore 255).



Original
High Threshold
Low Threshold



La scelta della **Threshold** (soglia) è fondamentale per stabilire quali sono i pixel tendenzialmente candidabili a diventare pixel di contorno, quindi una volta stabilita la soglia, tutti i pixel che si trovano al di sopra della soglia sono pixel di contorno, viceversa se si trovano sotto tale soglia non sono pixel di contorno. Infine, c'è una fascia grigia, che potrebbe oscillare tra una soglia troppo bassa o troppo alta.

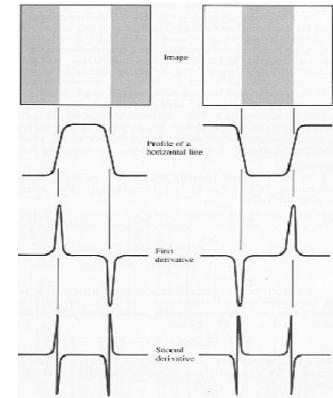
Con **Low Threshold** si individuano potenziali pixel di contorno anche pixel che hanno variazioni piuttosto piccole. Tali variazioni possono introdurre come potenziali pixel di controllo anche pixel che non lo sono. Questo tipo di soglia permette di individuare il profilo del pedone nero.

Con **High Threshold** si corre il rischio di ridurre troppo il numero di pixel che vengono etichettati come pixel di contorno, dando come risultato finale una quasi scomparsa totale della morfologia dell'oggetto. Il pedone ha perso tutta la parte centrale.

Il limite di questo approccio è che, in determinate condizioni, non ci assicura la possibilità di estrarre dei contorni chiusi.

Il fatto che la derivata prima e la derivata seconda del profilo siano significativamente diverse da 0 soltanto in corrispondenza alle transizioni costituisce la motivazione dell'uso di operatori derivativi per l'estrazione dei contorni. La **derivata prima** del profilo è **positiva** in corrispondenza di una transizione scuro-chiaro, **negativa** in corrispondenza di una transizione chiaro-scuro, **nulla** nelle zone a livello di grigio costante. La **derivata seconda** è **positiva** in prossimità di un contorno, dalla parte scura del contorno stesso, **negativa** dalla parte chiara del contorno, nulla nelle zone a livello di grigio costante, ed esibisce un passaggio per lo **zero o zero crossing** esattamente in corrispondenza delle transizioni. Ovvero, la **derivata prima indica la presenza di un massimo/minimo**, la **derivata seconda fornisce l'esatta posizione del massimo/minimo**, nella posizione in cui il rapporto incrementale smette di crescere per iniziare la decrescita.

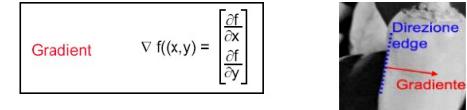
Riassumendo, il valore della derivata prima può essere utilizzato per determinare la presenza di contorni in una immagine. Gli zero crossing della derivata seconda ne possono consentire la precisa localizzazione. Il segno della derivata seconda permette di stabilire l'appartenenza di un pixel al versante scuro o al versante chiaro di un contorno. L'applicazione dei concetti precedentemente illustrati necessita tuttavia di alcune cautele, essenzialmente legate alla natura digitale delle immagini.



EDGE DETECTION – GRADIENTE:

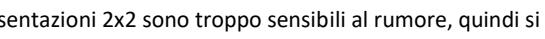
Il **gradiente** rappresenta uno degli strumenti attraverso il quale si può cercare di individuare le **zone di transizione**. Il gradiente di una funzione a due variabili viene calcolato come in figura. Del gradiente è possibile calcolare due informazioni: l'**entità del gradiente**, ovvero la grandezza della variazione, e la **direzione del gradiente**, ovvero in che direzione si muove il contorno. La direzione dell'edge è quella perpendicolare rispetto al gradiente. Il gradiente è uno strumento sofisticato che permette di calcolare due informazioni, se un pixel è etichettato come pixel di contorno sulla base della dimensione del gradiente, infatti più è grande la variazione, più il pixel ha probabilità elevate di essere etichettato come pixel di contorno. Il secondo aspetto è legato alla direzione del contorno che viene calcolata come perpendicolare rispetto alla direzione del gradiente.

La rappresentazione del gradiente può essere effettuata attraverso vari operatori, le rappresentazioni 2x2 sono troppo sensibili al rumore, quindi si usano le 3x3. Tutte quelle che si vedranno con linee orizzontali o verticali con valori zero sono rappresentazioni discrete del gradiente.



$$\text{Gradient Magnitude} \quad \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\text{Gradient Direction} \quad \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$



EDGE DETECTION – LAPLACIANO:

Il **Laplaciano** è una rappresentazione (implementazione) della **derivata seconda**, che a differenza del gradiente, è caratterizzato dall'avere un **pixel centrale positivo**, le **direzioni principali con valori negativi**, e gli **altri posti uguali a zero**. Si prendono i coefficienti della derivata seconda della funzione (x, y) rispetto a x , a due variabili. E questi poi costituiscono i coefficienti che si andranno a riprodurre all'interno della matrice. Questa rappresentazione del Laplaciano è la **rappresentazione discreta di una derivata seconda**.

Approximation of second derivative (horizontal):

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

convolution with: $[1 \ -2 \ 1]$

Laplacian Operator

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$$

convolution with:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Approximation of second derivative (vertical):

$$\text{convolution with: } \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

SHARPENING:

Un'operazione di **Sharpening** tende ad **evidenziare i dettagli fini**, attraverso un'operazione che prevede prima l'estrazione dei contorni e poi la sovrapposizione dei contorni con l'immagine originale. Quindi non si ottiene una immagine in bianco e nero ma, con le operazioni di Sharpening, dopo aver estratto i contorni essi vengono sovrapposti all'immagine, per renderli più evidenti.

Da come si evince in figura, si individua una immagine di partenza e successivamente si applica lo Sharpening utilizzando filtri differenziati con l'obiettivo però di non dare in output la semplice immagine dei contorni ma di sovrapporre i contorni ottenuti all'immagine originale.



$$M_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad M_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad M_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

CANNY EDGE DETECTOR:

Uno degli operatori maggiormente utilizzati per estrarre i contorni è l'operatore **edge detector Canny**, il quale si è dimostrato molto efficace. Questo operatore mette insieme tutto ciò che è stato detto finora, e sperimentalmente si è visto che è in grado di produrre risultati molto significativi. Questo perché il metodo di Canny ha un'elevata possibilità di produrre edge connessi, cioè **contorni chiusi** o piuttosto **continuativi** che sono uno degli obiettivi che si ricercano all'interno delle operazioni di **edge detection**. Quando si vuole effettuare l'estrazione di contorni e preservare la chiusura di essi, Canny fornisce questo tipo di prestazioni. L'**algoritmo di Canny** prevede:

1. *Smoothing gaussiano* dell'immagine, per eliminare le piccole transazioni;
2. Calcolo del **gradiente**;
3. Soppressione dei non-massimi in direzione ortogonale all'edge;
4. Selezione degli edge significativi mediante isteresi.

Con **Hysteresis**, ovvero una soglia intermedia dove vengono presi tutti i valori compresi tra una soglia alta e una soglia bassa. Due soglie, una bassa ed una alta, che vengono confrontate con il gradiente in ciascun punto. Se il valore del gradiente è:

- Inferiore alla soglia bassa, il punto è scartato;
- Superiore alla soglia alta, il punto è accettato come parte di un contorno;
- Compreso fra le due soglie, il punto è accettato solamente se contiguo ad un punto già precedentemente accettato.



1) Immagine originale – 2) Smoothing gaussiano – 3) Modulo del gradiente
4) Suppressione non-massimi – 5) Selezione edge – 6) Edge sovrapposti a immagine originale

3. TRASFORMATA DI FOURIER

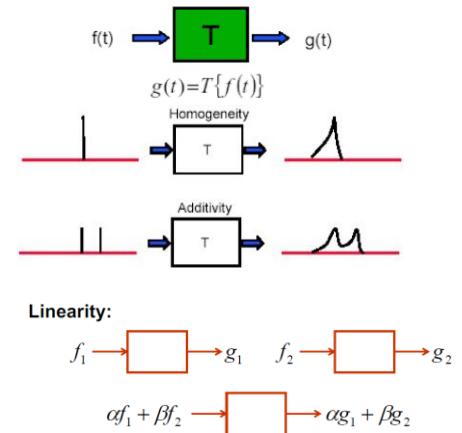
Il vantaggio dei *filtri spaziali* è che sono computazionalmente semplici, quelli visti sono lineari, mentre lo svantaggio è l'applicabilità, ovvero questi filtri vengono circoscritti a tutti quei casi in cui abbiamo una conoscenza dell'immagine, sappiamo quale applicare in base al contenuto e al rumore. L'obiettivo è costruire dei nuovi filtri che non agiscono direttamente sui pixel, ma agiscono su altre informazioni correlate con i pixel, ma a venti natura diversa. È chiaro che, quando si effettua una trasformazione di un'immagine da un dominio all'altro, si deve essere in grado di portare l'immagine da un dominio all'altro attraverso una trasformazione, ma fare anche l'***operazione inversa***, riportando l'immagine dal dominio nuovo al dominio dei pixel. Questa operazione può esser fatta con la ***trasformata di Fourier***. Intuitivamente si capisce che questo tipo di operazione comporta un ***aggravio computazionale***, per cui l'obiettivo è quello di capire se questo aggravio produce dei risultati qualitativamente migliori.

SISTEMA LINEARE:

Le trasformazioni lineare è una trasformazione che agisce su un *segnalet in ingresso* $f(t)$, che può essere associato ad un'immagine (funzioni discrete). Tale segnale entra in questo box verde e produce un *risultato* $g(t)$.

Per essere un sistema lineare, esso deve godere di alcune proprietà:

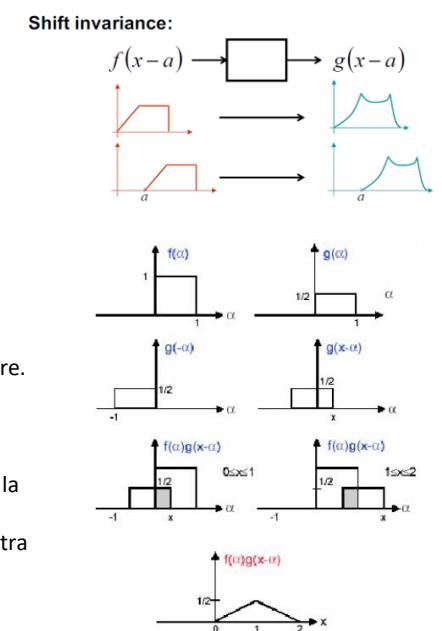
- Omogeneo se $T\{af[n]\}=aT\{f[n]\}$
- Associativo se $T\{f_1[n]+f_2[n]\}=T\{f_1[n]\}+T\{f_2[n]\}$



LINEAR SHIFT INVARIANT SYSTEMS (LSIS):

Un aspetto che si vuol introdurre nei sistemi lineare è la cosiddetta ***proprietà dello Shift Invariante***, cioè dei sistemi che sono invarianti al tempo. Un sistema è shift Invariant quando il suo comportamento nel tempo ***non cambia*** e l'unica variazione che si ottiene è la ***traslazione*** del risultato. Si può dire che un sistema gode dell'invarianza allo shift se ha un comportamento come nel seguente:

Se all'istante t il sistema riceve un segnale di questo tipo e produce un risultato simile, allora il sistema è Shift Invariant se all'istante $t = a$, con una traslazione temporale sull'asse delle ascisse, il sistema riceve un segnale analogo al precedente in input ma traslato temporalmente, allora produrrà come risultato lo stesso risultato precedente ma con una traslazione temporale. Se ciò avviene allora il sistema gode dell'invarianza allo shift. Quindi uguale comportamento ma traslato nel tempo.



CONVOLUZIONE IN CASO CONTINUO:

La ***convoluzione*** (simbolo *) tra due funzioni, nel caso continuo, si scrive nel seguente modo:

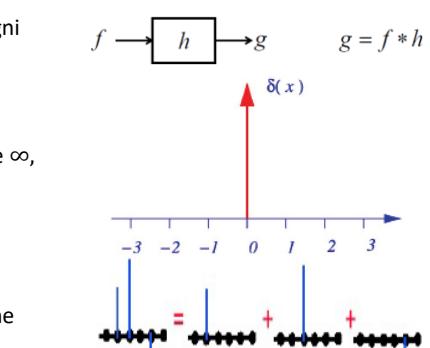
$$(f * g)(x) = \int_{-\infty}^{\infty} f(a)g(x-a) da$$

La presenza dell'integrale indica che il risultato di questa operazione, sarà un'area che si andrà a ritagliare.

Si mostrano degli esempi ->

La convoluzione fra f e g produce l'area che viene intersecata dalle due funzioni. Nell'ultimo grafico vi è la rappresentazione geometrica della convoluzione fra le due funzioni.

La convoluzione è un'operazione tra due funzioni di una variabile che consiste nell'integrare il prodotto tra la prima e la seconda traslata di un certo valore.



CONVOLUZIONE KERNEL - RISPOSTA ALL'IMPULSO:

Il ***Delta di Dirac*** è un impulso unitario, è la più semplice delle funzioni che permette di rappresentare ogni funzione in relazione al Delta di Dirac. Esso è una funzione:

$$\delta[n-k] = \begin{cases} 0 & \text{where } n \neq k \\ 1 & \text{where } n = k \end{cases}$$

La funzione ha le caratteristiche che vale 0 tra $-\infty$ e ∞ , tranne nel punto 0 dove il valore della funzione è ∞ , e l'integrale della funzione tra $-\infty$ e ∞ è uguale a 1.

Se viene considerata una funzione di questo tipo, si può rappresentare una qualsiasi funzione come:

$$f[n] = \sum_{j=-\infty}^{\infty} f[j] \delta[n-j]$$

La funzione può essere decomposta in somma lineare di impulsi scalati e shiftati, ciascuno dei quali viene intercettato dal Delta di Dirac.

SISTEMA LINEARE SHIFT INVARIANTE E CONVOLUZIONE:

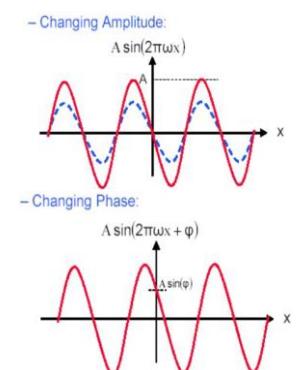
L'obiettivo è la realizzazione di un operatore T (trasformazione) che viene applicata non più direttamente ai pixel ma ad una funzione trigonometrica (sinusoidale) che venga trasformata da T agendo su ***ampiezza*** e ***fase***, non alterando la frequenza. Quindi la funzione entra con una frequenza omega ed esce con la stessa frequenza, ma potrà essere amplificata, ovvero essere aumentata o contratta per effetto di un coefficiente moltiplicativo, ma anche eventualmente sfasata, ovvero shiftata temporalmente mantenendo però sempre la stessa frequenza.

Lavorando sui pixel, T è stato costruito con le matrici e i pesi relativi, ottenendo T diversi tra loro. In questo caso si vuol costruire T affinché agisca sulle funzioni trigonometriche.

Quindi la risposta di un sistema lineare shift invariante ad una funzione seno è una copia scalata e shiftata della funzione seno con la stessa frequenza.

Nel ***dominio spaziale*** si fa entrare un impulso e il sistema risponde con una funzione trasformata.

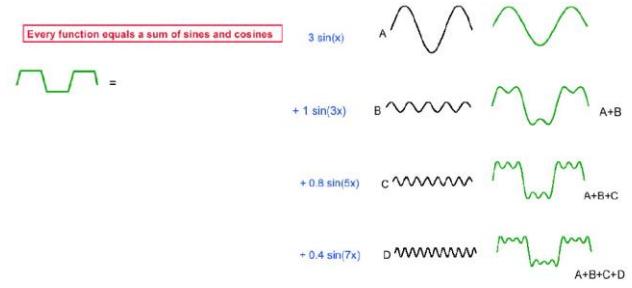
Nel ***dominio delle frequenze*** si fa entrare un'onda sinusoidale e il sistema è in grado di modificare la sua ampiezza.



TRASFORMATA DI FOURIER:

Fourier affermò che una *funzione periodica*, può essere scritta sempre sottoforma di somma pesata di *funzione seno e coseno*, prese a frequenze diverse.

Come si può notare, le varie funzioni sommate tra loro riescono ad approssimare la funzione di partenza. Con infinite frequenze, aumenta maggiormente la frequenza che si avvicina alla funzione di partenza. Quindi, infiniti punti nel dominio spazio-tempo corrispondono a infinite frequenze nel dominio delle frequenze. **Numero finito di pixel nel dominio delle frequenze, finite frequenze nel dominio delle frequenze.**



ANALISI DELLE FREQUENZE:

Si formalizza ora la Trasformata di Fourier, nella rappresentazione formale:

$$f(x) = \sum_{\omega} F(\omega) e^{i 2 \pi \omega x}$$

Or

$$f(x) = \int F(\omega) e^{i 2 \pi \omega x} d\omega$$

La funzione $f(x)$ è la funzione nello spazio-tempo (nel contesto dei pixel è dominio di pixel). Quindi tale funzione è rappresentabile come la sommatoria di ω , dove essa rappresenta le frequenze di $F(\omega)$ che è l'ampiezza (ovvero il peso) moltiplicato per $e^{i 2 \pi \omega x}$. Quindi per ogni frequenza ω , si assegna un peso stabilito. Questa trasformata di Fourier permette di prendere un segnale e rappresentarlo in termini di funzioni sinusoidali a frequenza ed ampiezza variabile, cioè pesate. Ciò può essere scritto sotto forma di sommatoria oppure di integrale.

Per essere completi dal punto di vista della **trasformata di Fourier** si necessita di una soluzione che dia la possibilità di passare dal dominio spazio-tempo al dominio delle frequenze e viceversa. Il primo passaggio da effettuare è quello di costruire le frequenze partendo dalla funzione. $F(\omega)$ rappresenta l'insieme delle ampiezze che si assegnano alle varie frequenze. Questa trasformazione porta dal dominio spazio-tempo al dominio delle frequenze. Quando si vuol fare l'operazione inversa si applica la **funzione inversa**. Se vi è un ritorno dal dominio delle frequenze al dominio spazio-tempo, caratterizzato dall'assenza di modifiche è assolutamente indolore, ovvero non si perdono operazioni. **Se l'informazione nel dominio spazio-tempo è composta da n pixel, si avranno n frequenze nel dominio delle frequenze e viceversa.**

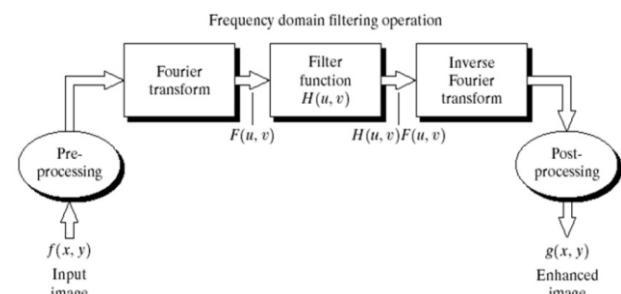
Si parte da una immagine nel dominio dei pixel $f(x, y)$, dove x e y corrispondono alle coordinate della posizione del pixel ed f corrisponde al valore di luminosità associato a questa posizione. L'immagine successivamente viene processata mediante una trasformata di Fourier e quindi si vanno a rappresentare i pixel in termini di frequenze, come nell'enunciato precedente. Ora l'idea è costruire un filtro nel dominio delle frequenze che può essere applicato ad F attraverso una operazione di moltiplicazione esattamente come fatto nei filtri spaziali. Fatto ciò, si applica la trasformata inversa per ottenere $g(x, y)$. Quindi invece che applicare la trasformazione direttamente ad f , si applica la trasformazione ad una rappresentazione nel dominio delle frequenze di f .

The **inverse Fourier Transform** composes a signal $f(x)$ given $F(\omega)$:

$$f(x) = \int F(\omega) e^{i 2 \pi \omega x} d\omega$$

The **Fourier Transform** finds the $F(\omega)$ given the signal $f(x)$:

$$F(\omega) = \int f(x) e^{-i 2 \pi \omega x} dx$$



TRASFORMATA DI FOURIER DISCRETA:

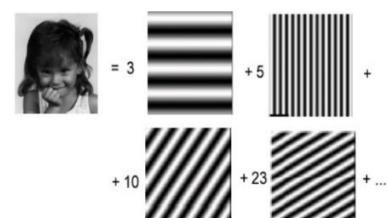
The **Discrete Fourier Transform** (DFT) is defined as:

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

The **Inverse Discrete Fourier Transform** (IDFT) is defined as:

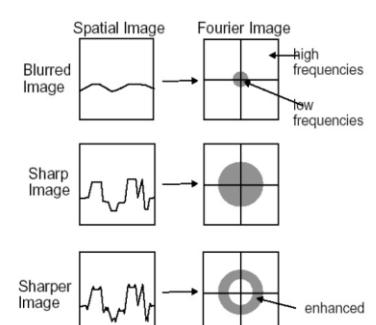
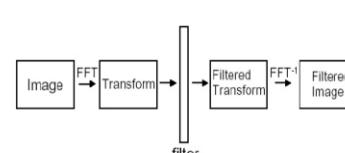
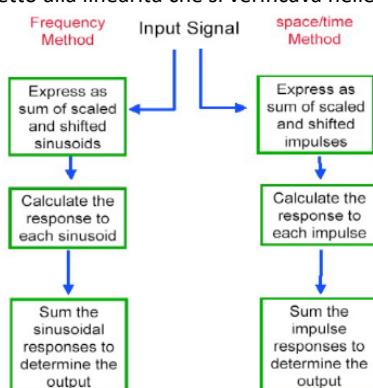
$$\hat{f}(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{2\pi i ux}{N}} \quad x = 0, 1, 2, \dots, N-1$$

Quando si applica la trasformata di Fourier si cerca un certo numero di funzioni seno e coseno. Come si vede in figura, **dalla combinazione tra le varie funzioni si ottiene la funzione di partenza**.



DOMINIO SPAZIALE VS FREQUENZA:

Il problema iniziale è filtrare l'immagine, quindi in qualche modo bisogna risolvere un problema nel dominio spazio-tempo, in tale dominio *vi sono delle difficoltà* perché i pixel hanno una natura molto diversificata, quindi ciò può comportare dei problemi. Ed è per questo che si fa ricorso all'utilizzo di soglie che non sempre possono essere precise. Potrebbe essere più funzionale applicare la trasformata di Fourier per tentare di risolvere il problema operando in un dominio differente, ovvero il **dominio delle frequenze**, dove dovrebbe essere più semplice rilevare qualche soluzione per il problema in questione. Si dovrebbe verificare che, filtrare una immagine attraverso la propria rappresentazione nel dominio delle frequenze, risulta essere più semplice e questa soluzione, attraverso la trasformata di Fourier inversa, che riporta al dominio spazio-tempo comporterebbe avere la soluzione del problema originale. La trasformata di Fourier, applicata ad un'immagine NxN, computazionalmente viene a costare **$O(N^2)$** , quindi un aumento di complessità rispetto alla linearità che si verificava nelle trasformate lineari. La **Fast Fourier Transform** costa **$O(n \log n)$** .

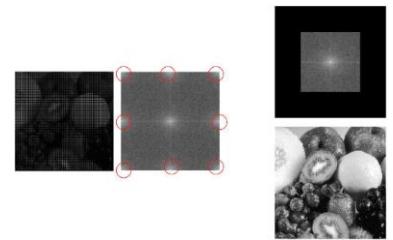


$$G(\omega) = F(\omega)H(\omega)$$

$$g(x) = f(x) * h(x)$$

ESEMPIO DI FILTRAGGIO CON FOURIER:

In figura, vi è una immagine con rumore. È molto difficile rimuovere il rumore con i filtri enunciati in precedenza, di fianco vi è la rappresentazione della trasformata di Fourier, dove viene rappresentato lo **spettro di Fourier**, ovvero la rappresentazione delle frequenze (o meglio delle ampiezze), che si sono ottenute nella trasformata. Le ampiezze nello spettro di Fourier si possono interpretare come: tendono ad essere **più alte** in prossimità delle **basse frequenze**, e tendono progressivamente ad **azzerarsi** man mano che le frequenze **aumentano**. I bagliori cerchiati in rosso indicano delle ampiezze significative nelle alte frequenze, il che è atipico per le immagini e ciò significa che verosimilmente l'immagine è corrotta dalla presenza di rumore, e quindi tali aree vanno eliminate (moltiplicandole per 0 quelle frequenze).

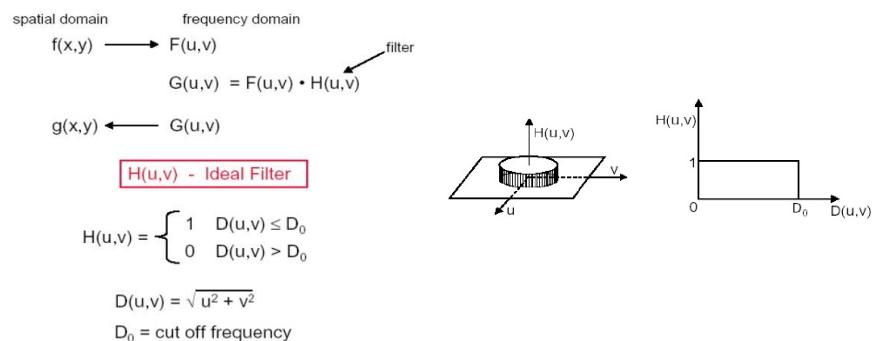


Per fare tale operazione vanno azzerate le ampiezze che riguardano le alte frequenze, ovvero una **moltiplicazione per zero** per tutti i coefficienti delle alte frequenze, e una **moltiplicazione per uno** per le frequenze che si trovano nella parte bassa e media. Ottenendo uno spettro di Fourier come in alto a destra, e il risultato dell'immagine, operando l'anti trasformata di Fourier, è una immagine senza rumore e qualitativamente migliore. Con un filtro spaziale non si sarebbe mai ottenuto tale risultato. La trasformata di Fourier rappresenta attraverso il suo spettro, la **distribuzione delle ampiezze** delle varie funzioni trigonometriche utilizzate, una immagine senza rumore o con pochi dettagli, avrà un bagliore localizzato vicino allo zero (centro), infatti vi sono poche frequenze basse e quasi nulla di alte frequenze. Se l'immagine presenta molti contorni e molti dettagli, lo spettro di Fourier avrà un bagliore molto più ampio, che sta ad indicare che ci sono molte frequenze con ampiezze più significative.

LOW PASS FILTER:

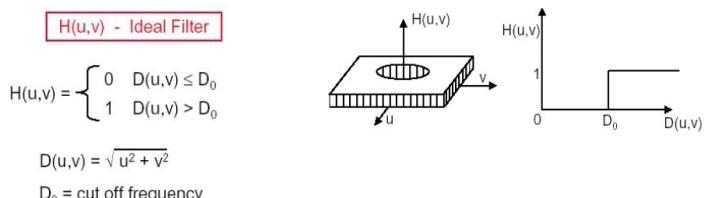
Quando si costruisce un filtro, bisogna decidere quali frequenze conservare e quali no. La prima cosa fattibile è la costruzione di un **filtro passa basso**, ovvero un filtro che lascia **transitare** solamente **basse frequenze**.

Tale filtro viene costruito sulla base dell'individuazione delle frequenze che si vogliono conservare e per queste frequenze si pensa ad una moltiplicazione per uno, e per quelle che si vogliono tagliare quindi una moltiplicazione per zero. Il filtro diventa un moltiplicatore, attuato sulle frequenze. D_0 rappresenta la **frequenza di taglio**, ovvero quella rispetto alla quale si stabilisce una soglia.



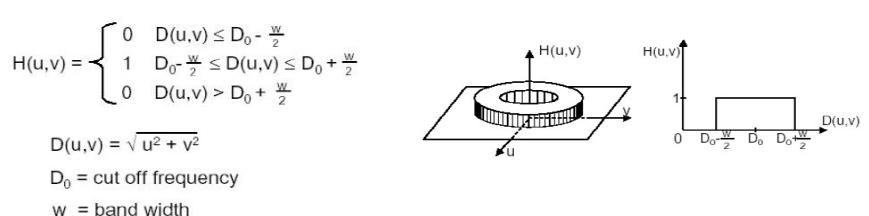
HIGH PASS (SHARPENING) FILTER:

Il filtro passa alto sarà opposto al precedente, esso mette in evidenza i dettagli.



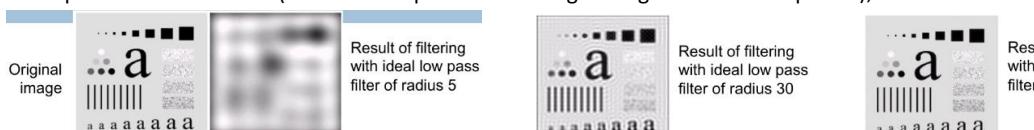
BAND PASS FILTER:

Prende una parte delle basse frequenze e una parte delle alte frequenze. Quando si aggiunge del rumore, esso si annida nella alte frequenze, ma mai nelle basse frequenze. Se si moltiplica per zero le frequenze del rumore, esso scomparirà senza intaccare la qualità dell'immagine.



FREQUENCY BANDS:

Ragionando in termini di **energia** del segnale (contenuto informativo presente nell'immagine), quando si taglano le frequenze alte rimane circa il 90% delle informazioni dell'immagine. Se si alza questa soglia, e si prendono anche altre frequenze, le informazioni dell'immagine aumenta. Questo significa che anche prendendo un numero basso di frequenze si riesce a conservare gran parte dell'informazione significativa dell'immagine. L'occhio è molto sensibile alle frequenze medie e basse (informazioni più vicine all'origine degli assi delle frequenze), e meno sensibile alle alte frequenze.



SISTEMA LINEARE E RISULTATI:

Si analizzano le analogie tra il **dominio delle frequenze** e il **dominio spaziale** (pixel).

Il segnale iniziale corrisponde all'immagine e per essa vi è la corrispondente trasformata di Fourier. L'output prodotto nel dominio delle frequenze è un segnale G che opportunamente trasformato con l'anti-trasformata di Fourier diventa g . Per costruire il filtro si individuano quali frequenze tagliare e quali conservare, sulla base di tale scelta viene effettuata la moltiplicazione $G = FH$ e si ottiene G che con l'opportuna anti-trasformata si tramuta in g . L'analogia tra i due domini indica che si potrebbe effettuare la stessa operazione nel dominio spaziale, ma non effettuando una moltiplicazione ma bensì una convoluzione tra f ed h , dove h dovrebbe essere ottenuto come anti-trasformata di Fourier di H . Però mentre H è facile da costruire nel dominio delle frequenze (vettore di 1 e 0 per conservare o tagliare), nel dominio spaziale la costruzione di h è molto complessa, infatti non si ha possibilità di costruirlo, se non facendo l'anti trasformata di H , in più la relazione tra i due domini non implica la stessa operazione, infatti da un lato c'è la **convoluzione** e dall'altro una **moltiplicazione**.

TEOREMA DI CONVOLUZIONE:

Il **teorema di convoluzione** stabilisce che quando abbiamo due domini in relazione, ovvero dominio spazio-tempo e frequenze, una moltiplicazione tra due funzioni in un dominio implica una convoluzione nell'altro dominio, e viceversa. Questo permette di stabilire una relazione tra le operazioni nei diversi domini.

	Spatial Domain	Frequency Domain
Input	f	F
Output	g	G
Impulse Response	h	
Freq. Response		H
Relationship	$g = f * h$	$G = F \cdot H$

CAMPIONAMENTO:

È il processo attraverso il quale si trasforma un segnale **continuo**, quindi **analogico**, in un segnale **discreto**. Normalmente avendo un segnale continuo nel dominio spazio-tempo, si può individuare un campionamento a **passo T**, che tende ad individuare un **sottoinsieme di punti** (un insieme discreto e finito di punti) che dovrebbero approssimare il segnale di partenza. Nel dominio spazio-tempo determinare con certezza la dimensione del passo di campionamento risulta essere piuttosto complicato, sapendo che la qualità del segnale discreto ricostruito è condizionata dal passo di campionamento. In genere più è ampio il passo di campionamento minore è la qualità del segnale discreto.

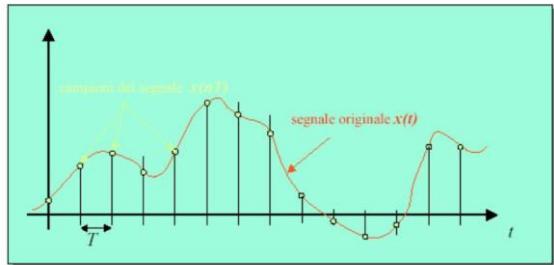
Il **teorema del campionamento** permette di determinare un passo di campionamento attraverso il quale si può quantificare la quantità di informazione che si andrà a **perdere**, stabilendo anche la **qualità** dell'informazione che verrà persa. Per fare questo si ha bisogno di effettuare un passaggio per un dominio differenziato rispetto a quello spazio-tempo (che nel caso delle immagini corrisponde a quello dei pixel). Si dovrà quindi passare alla rappresentazione in termini di trasformata di Fourier.

CAMPIONAMENTO DI UNA IMMAGINE:

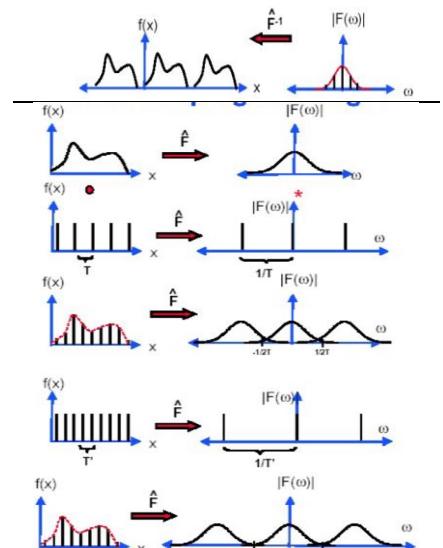
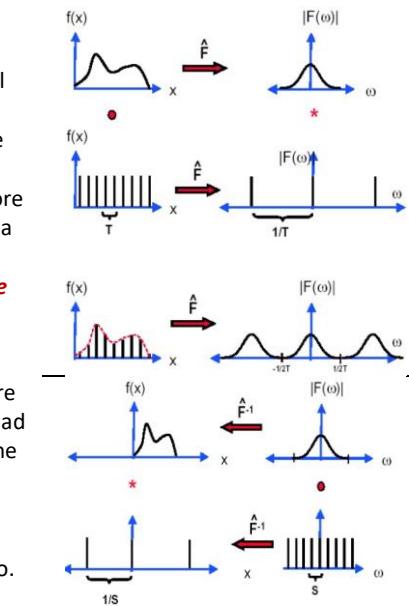
Si immagini di avere un segnale nel dominio spazio-tempo, tutti i grafici a sinistra (delle immagini a destra) sono grafici del dominio spazio-tempo, a destra il dominio delle frequenze. Si sa che un segnale qualsiasi nel dominio spazio-tempo può essere trasformato, attraverso Fourier, in una sua rappresentazione, ovvero lo **spettro di Fourier**, esso è la rappresentazione in valore assoluto dei coefficienti assegnati alle armoniche. Se nel dominio spazio-tempo si volesse effettuare un campionamento, si dovrebbero produrre una serie di impulsi unitari che dovrebbero intercettare il segnale al passo t . Più fitta sarà questa serie di impulsi maggiore sarà la probabilità di ricostruire in maniera fedele il segnale. Se si effettua la trasformata di Fourier su questa serie di impulsi a passo t si otterrà nel dominio delle frequenze, quindi come spettro di Fourier, una serie di impulsi a passo $1/t$. **Più sarà piccolo il passo di campionamento nel dominio spazio-tempo, più sarà grande l'intervallo nel dominio delle frequenze.**

Sapendo che tra i due domini, per il **teorema di convoluzione**, esiste una corrispondenza specifica, effettuare un'operazione di moltiplicazione tra questa serie di impulsi e il segnale nel dominio spazio-tempo, equivale ad effettuare la convoluzione tra il segnale in valore assoluto e la frequenza di campionamento. La convoluzione produce **repliche del segnale originario**, avendo una serie di gaussiane replicate lungo il dominio di riferimento, questo corrisponderà fondamentalmente a quello ottenuto, in termini di moltiplicazione nel dominio spazio-tempo. Le operazioni effettuate verificano la corrispondenza tra l'effettuare un campionamento nel dominio spazio-tempo e mostrare i risultati per le operazioni effettuate, in tale dominio.

Campionare i segnali (discretizzare nel tempo)



- T è detto periodo (o passo) di campionamento
- $f_c = 1/T$ è detta frequenza di campionamento



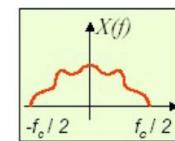
Si mostra ora cosa succederebbe se si volesse effettuare un campionamento nel dominio spazio-tempo o nel dominio delle frequenze. Nel dominio spazio-tempo dovrebbe essere trovato un passo di campionamento t relativamente piccolo affinché esso garantisca la conservazione del maggior numero di punti sul segnale originario. Più t aumenta, più il numero di punti presi in considerazione diminuisce. Se si effettua la stessa considerazione nel dominio delle frequenze, poiché la trasformata di Fourier del segnale t , produce una serie di campioni a passo $1/t$, nel dominio delle frequenze, più il passo di campionamento nel dominio spazio-tempo risulta essere piccolo, più i segnali unitari risultano essere distanziati. Questo distanziamento, in termini di convoluzione, ha effetto su quanto le armoniche risultano essere intersecate. Più t è piccolo, più le armoniche saranno **disgiunte**, più t è grande più le armoniche tenderanno a **sovraporsi**. Quando le armoniche sono molte sovrapposte, una parte delle informazioni vanno tra loro a **confondersi**, ovvero le informazioni che risiedono nelle **alte frequenze** come **contorni** e **dettagli**, risultano essere ambigue (un punto corrisponde a due punti di due armoniche), quindi l'ideale sarebbe poter ottenere delle curve che non vanno ad intersecarsi ma che siano **contigue**. Per stabilire il giusto passo di campionamento, bisogna stabilire un **passo di campionamento**: $\frac{1}{T} > 2\omega_{Max}$, dove ω_{Max} è la frequenza massima del segnale.

Il **teorema di Nyquist** permette di ottenere delle curve che *siano contigue senza sovrapposizioni*, riducendo anche il problema dell'**aliasing**, ovvero la perdita delle informazioni che risiedono nelle alte frequenze e quindi la non perdita di dettagli e contorni.

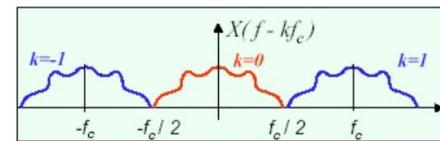
Come mostrato in figura, questa è la situazione ideale dove nel dominio delle frequenze si dovrebbe ottenere **repliche della rappresentazione** dello spettro di Fourier che siano contigue, scegliendo come frequenza di campionamento quella che consente la **non sovrapposizione** nelle alte frequenze. Il teorema di Nyquist permette di individuare un passo di campionamento che evita la perdita di informazioni, stabilendo quale valore di campionamento utilizzare nel dominio delle frequenze, tale da ottenere un numero discreto di punti che conservino tutte le informazioni originarie del segnale.

Questo è vero solo se il segnale possiede banda limitata ovvero se esiste un ω_{max} . Se quest'ultimo non esiste, ovvero le gaussiane non convergono verso zero ma si muovono all'infinito senza mai toccare l'asse delle ascisse, allora è chiaro che non si potrebbe avere una ricostruzione perfetta del segnale nel dominio discreto, e quindi si andrà a dimensionare quali informazioni devono essere tagliate (in genere quelle nelle altissime frequenze). Il processo di campionamento nel dominio delle frequenze permette di stabilire che se il segnale ha banda limitata, si può campionare il segnale senza perdita di informazioni, se viceversa il segnale non è a banda limitata, quindi le frequenze tendono all'infinito, allora si possono quantificare le informazioni che andranno perse. Questo permette di stabilire quante informazioni perdere e la qualità di esse.

$$x(t) \xrightarrow{\text{TF}} X(f)$$



$$x(t) \exp(j2\pi f_c t) \xrightarrow{\text{TF}} X(f - kf_c)$$



4. INTRO ALLA BIOMETRIA

Il problema della biometria, di una certa rilevanza, corrisponde al verificare se l'**identità dichiarata** corrisponde all'**identità reale**.

AUTENTICAZIONE DI UN SOGGETTO:

Esistono vari processi di autenticazione di un soggetto. Un **processo di associazione** dell'identità ad un soggetto avviene attraverso un processo di **verifica o riconoscimento**. Una persona ne riconosce un'altra attraverso supporti di identificazione (carta d'identità o codice fiscale), o per conoscenza diretta (ambito familiare). Invece, un sistema informatico riconosce una persona utilizzando sistemi di identificazione (smart card, biometriche, ecc...). Queste metodologie presentano dei limiti come, ad esempio, in presenza di un PIN (login e password), è possibile dimenticarlo oppure essere carpito da un impostore. Anche nel caso delle chiavi fisiche (smart card) che si possono rompere o perdere. Questi sistemi non consentono la distinzione tra il reale possessore e un impostore. Per superare i limiti dei sistemi di autenticazione attualmente in uso, si usano:

- **Token** (qualcosa che si possiede), carte magnetiche o Smart, chiavi, passaporto, ecc...
- **Password** (qualcosa che si conosce), password, PIN, ecc...
- **Biometria** (ciò che si è), una caratteristica fisica o comportamentale non associabile a nessun'altra persona.

Si deve superare il meccanismo di autenticazione tradizionale basato sul concetto che l'utente possiede, spostando l'attenzione su qualcosa che "l'utente è effettivamente".

BIOMETRIA:

Dal greco bios (vita) e metros (misura), la **biometria** è basata sul riconoscimento automatico di una persona sulla base di caratteristiche discriminanti. La biometria è la disciplina che si occupa di misurare tratti fisiologici e comportamentali degli esseri viventi, offrendone una rappresentazione quantitativa attraverso modelli matematici e statistici per costruirne un template ed effettuare eventuali confronti.

Le **tecnologie biometriche** costituiscono metodi automatizzati di **verifica o riconoscimento** dell'identità di una persona, basati sul rilevamento di una o più **caratteristiche fisiologiche o comportamentali** dell'individuo, oppure il confronto con una immagine precedentemente acquisita.

I principi portanti della biometria sono:

- Ogni persona è **unica**, dove alcune biometrie sono più specifiche di altre;
- Individuazione delle **caratteristiche somatiche** che rendono unico un individuo;
- Metodologie per la **misurazione e quantificazione** di tali caratteristiche;
- **Classificazione** degli individui sulla base delle misure effettuate.

LIVELLI DI AUTENTICAZIONE:

I **livelli di autenticazione** normalmente utilizzati, dipendono dal livello di sicurezza che si intende raggiungere. Vi sono dei livelli di sicurezza in cui il livello di autenticazione deve essere molto forte.

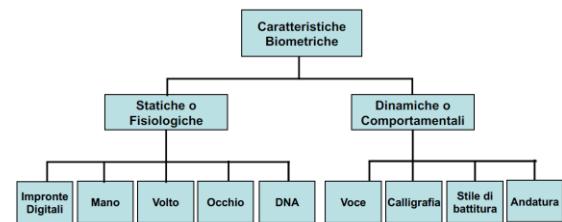
Partendo dal basso, si individuano i livelli di sicurezza meno forti, come ad esempio quelli legati a qualcosa che si conosce come una password o PIN. Un livello superiore è il cosiddetto Token, basato su qualcosa che si deve avere come una chiave o un badge, quasi sempre legate a qualcosa che si conosce per aumentarne il livello di sicurezza. Infine, vi è un'informazione biometrica, fisiologica o comportamentale, quindi qualcosa che realmente "si è" che possa essere eventualmente abbinata alle informazioni appena descritte. L'incrocio di questi tre livelli di sicurezza produce un livello di sicurezza molto elevato. Le altre combinazioni tra di esse rendono comunque alto il livello di sicurezza.



CARATTERISTICHE BIOMETRICHE:

Parlando di **caratteristiche biometriche** è possibile effettuare una classificazione:

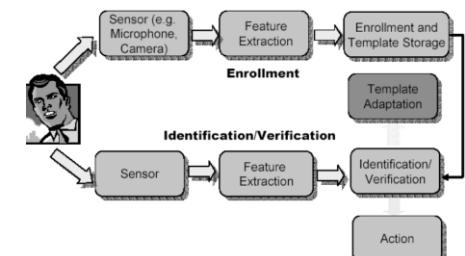
- **Statiche o Fisiologiche**, hanno una loro stabilità nel tempo ma soprattutto non dipendono dal comportamento della persona e non possono essere variate sulla base di aspetti emotivi, ad esempio, impronta digitale, occhio, volto, ecc...;
- **Dinamiche o Comportamentali**, dipendono fortemente dagli aspetti emotivi della persona, e possono essere anche variabili, avendo un impatto sulla base di aspetti emotivi o legati alla salute o stile di vita, ad esempio, voce, calligrafia, andatura, ecc....



Per poter utilizzare una biometria nel contesto forense, è necessario che esse rientrino in una normativa (**Standard X9.84 - 2003**), affinché esse siano legalmente valide. L'utilizzo della biometria comporta una serie di problemi, ovvero utilizzato come strumento di riconoscimento e tracciamento di una persona, può intaccare alcuni **aspetti etici** e della **privacy**.

ARCHITETTURA DI UN SISTEMA BIOMETRICO:

Un sensore che sia in grado di trasformare l'informazione biometrica in un segnale digitale. Una volta digitalizzato tale dato, vengono estratte delle caratteristiche, che dipendono dal tipo di biometria utilizzata. Tali caratteristiche saranno la base della creazione del modello matematico (**template**). Questo modello diviene il template rispetto al quale, successivamente si faranno operazioni di **verifica e riconoscimento**. Con la creazione del template, si è in grado di poter riconoscere un determinato individuo, riacquisendolo in maniera analoga a come appena descritto, e una volta creato il template temporaneo verificare se esso è presente nell'archivio per produrre un risultato.



REGISTRAZIONE DI UN SOGGETTO (ENROLLMENT):

Si associa un insieme di caratteristiche all'identità di un soggetto, raccogliendo dei dati ed **estraendo delle caratteristiche**. Il modello estratto (**template**) è memorizzato in un database o su un supporto portatile (**smart card**). Questo processo può essere effettuato singolarmente o in blocco (**batch Enrollment**). Per una biometria possono essere creati diversi modelli matematici (template), con diversi algoritmi.

VERIFICA E RICONOSCIMENTO:

Non sempre una stessa biometria può essere utilizzata sia per verifica che per fare riconoscimento, ma dipende dalle condizioni, una delle quali è proprio scegliere se svolgere **Verifica o Riconoscimento**. Nel caso delle **Verifica** si parla di confronto **Uno a Uno**, dove un individuo fornisce la propria identità (mediante carta d'identità, codice utente, ecc...) in un qualsiasi modo, e il sistema una volta che è stata acquisita l'identità, effettua un match 1 a 1, acquisendo il template della persona in real time, confrontandolo con quello memorizzato, e confermando o negando l'identità dichiarata da un individuo. Viceversa, nella fase di **Riconoscimento** dove si verifica un confronto **Uno a Molti**, ovvero quando l'utente non fornisce la propria identità (telecamera aeroportuale). In questo caso si acquisisce il dato biometrico e lo si confronta con quelli memorizzati in vari storage, e si stabilisce l'identità di un soggetto a partire da un insieme di persone registrate.

FORMAZIONE DELLE BIOMETRICHE:

Alcune biometriche risultano essere più discriminanti di altre. Durante la fase di **angiogenesi**, nel grembo materno impatta maggiormente sulla formazione di due delle biometriche, più discriminanti e stabili nel corso della vita di un utente, ovvero **impronta digitale** e **iride**.

Nella prima tabella (più cerchi indicano una corrispondenza forte), vengono analizzate alcune biometriche, e per ognuna di esse vengono individuati tre possibili fattori di caratterizzazione, **genotipico**, **randonotipico** (o **fenotipico**) e **comportamentale**. Concentrandosi sui primi due si può evincere che le componenti **genotipiche** vengono trasmesse direttamente attraverso il genoma dei genitori. La componente **randonotipica** non è condivisa con nessuno ed è strettamente legata all'utente. Quanto più è forte quest'ultima componente tanto più si riesce a differenziare una persona, anche rispetto ai parenti. L'altro elemento che si analizza è legato alla durata nel tempo di una biometria, nella seconda tabella. Se una biometria risulta essere molto variabile, non è pratico utilizzarla per lunghi periodi di tempo.

Biometric characteristic	genotypic*	randonotypic*	behavioral**	Biometric Trait	Permanence over time
Fingerprint (only minutiae)	0	ooo	o	Fingerprint (Minutia)	oooooo
Signature (dynamic)	oo	o	ooo	Signature(dynamic)	oooo
Facial geometry	ooo	o	o	Facial structure	oooooo
Iris pattern	o	ooo	o	Iris pattern	oooooooooooo
Retina (Vein structure)	o	ooo	o	Retina	oooooooooooo
Hand geometry	ooo	o	o	Hand geometry	oooooooooooo
Finger geometry	ooo	o	o	Finger geometry	oooooooooooo
Vein structure of the hand	o	ooo	o	Vein structure of the back of the hand	oooooo
Ear form	ooo	o	o	Ear form	oooooo
Voice (Tone)	ooo	o	oo	Voice (Tone)	ooo
DNA	ooo	o	o	DNA	oooooooooooo
Odor	ooo	o	o	Odor	oooooo ?
Keyboard Strokes	o	o	ooo	Keyboard strokes	oooo

La tabella indica la robustezza delle biometrie, dove il numero di cerchi è relazionato alla robustezza di tale biometria, ovvero più cerchi più robustezza. Quindi una biometria se è stata fortemente condizionata da elementi randotipici risulta essere più discriminante.

STORIA DELLA BIOMETRIA:

Bertillon fu il primo ad esasperare il concetto di classificazione già fondato dagli egiziani in precedenza.

Gli egiziani usavano l'altezza di una persona per identificarla durante il pagamento del salario. Il sistema Bertillon (1853-1914) creò un nuovo sistema per identificare i criminali. I principi del suo sistema era che l'ossatura umana non si modifica più partire dai 20 anni e ogni scheletro è diverso, ma anche da misure antropometriche (lunghezza braccio e altezza e larghezza testa) e anche descrizioni generali del corpo, caratteristiche mentali e morali. **Bertillon** creò una scheda per classificare i pregiudicati, dove essi venivano classificati sulla base di alcune caratteristiche biometriche descritte. Questa scheda riportava dati che avevano una certa affidabilità, benché nascessero da misurazioni piuttosto variabili dati gli strumenti utilizzati.

Il **Bertillionage** (sistema di classificazione) composto da queste schede che andavano a comporre un archivio, fu adottato in varie parti del mondo. Esso durò per qualche anno fin quando non si verificò un caso di omonimia, che portò questo sistema di classificazione al fallimento data la comprovata poca affidabilità. Il sistema fallì in quanto furono registrati due detenuti William West e Will West con misure antropometriche simili.

Il **sistema di Bertillon** fu sostituito, anche perché in quel periodo fu messo a punto un sistema di classificazione delle persone molto più affidabile, basato su caratteristiche biometriche delle impronte. Vennero scoperte nella struttura tridimensionale delle impronte che esistevano delle caratteristiche molto piccole non sempre visibili (le **minuzie**) che rendevano la possibilità di confusione tra due persone molto bassa.

PROPRIETÀ DI UNA CARATTERISTICA BIOMETRICA:

- Universalità**: quando si sceglie una biometria, ogni persona deve possedere tale biometria;
- Distintività (Unicità)**: due persone devono essere sufficientemente distinguibili in base a tale biometria;
- Permanenza (Stabilità)**: la biometria deve rimanere invariante (rispetto ai termini di confronto) nel tempo;
- Misurabilità (Collezionabile)**: essa deve essere misurabile quantitativamente;
- Performance (Efficacia/Efficienza)**: l'acquisizione, l'estrazione delle caratteristiche e il confronto delle entità non devono essere costose in termini di tempo, memoria ed efficienza del sistema;
- Accettabilità**: la procedura di acquisizione deve essere tollerata da una vasta porzione della popolazione (utenti a cui viene sottoposta la biometria);
- Eludibilità**: la probabilità che il sistema venga raggiunto o indotto in errore deve essere minimizzata.

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke Dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retina	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

TIPOLOGIA DI UTENTE E PROCEDURA:

Nella definizione della biometria più adatta ad un determinato contesto operativo, bisogna considerare se l'utente è:

- Cooperativo**: è interesse dell'utente che il sistema lo riconosca (un impostore cerca di farsi riconoscere).
- Non-Cooperativo**: l'utente è indifferente al processo di riconoscimento (un impostore cerca di non farsi riconoscere).

Altra possibile considerazione riguarda l'effettuare il riconoscimento con una procedura:

- On-line**: l'utente è in attesa finché il sistema non risponde (esempio cellulare, tempi brevi di risposta).
- Off-line**: l'utente viene acquisito ma non attende una risposta immediata (esempio telecamera di sorveglianza).

MISURA DELLE PRESTAZIONI:

Le prestazioni di un sistema biometrico possono essere misurate con diverse modalità, innanzitutto comprendendo il comportamento di tale sistema biometrico. Tendenzialmente ci possono essere quattro possibili casi:

- L'identità **corrisponde** ed il soggetto è **accettato**;
- L'identità **corrisponde** ma il soggetto è **respinto** (errore);
- Un **impostore è accettato** (errore);
- Un **impostore è rifiutato**.

Non è possibile riconoscere un individuo senza ammettere una **tolleranza d'errore**. La soglia tra template e riconoscimento è cruciale e dipende dall'applicazione. Per misurare le prestazioni si può considerare 2 tipi di errori:

- Una soglia troppo bassa causa molti errori di **Tipo I** (rifiuto di soggetti registrati) – **FRR (False Recognition Rate)**;
- Una soglia troppo alta causa molti errori di **Tipo II** (accettazione di impostori) – **FA (False Acceptance Rate)**.

La scelta più utilizzata è **Equal Error Rate (EER)**: FAR = FRR.

Si pensi ad un sistema biometrico dove bisogna stabilire la giusta soglia di tolleranza. In figura sono costruiti dei grafici, dove sull'asse delle ascisse viene posta la soglia di tolleranza, e sull'asse delle ordinate viene posto l'errore commesso (essi dipendono dalla soglia di tolleranza). In figura viene mostrato FAR che, quando la soglia di tolleranza risulta essere piccola, l'errore commesso è molto basso, ma man mano che la soglia di tolleranza aumenta, ammettendo un errore maggiore, la probabilità che un impostore possiede nel farsi accettare cresce in maniera esponenziale.

L'FRR si comporta in maniera opposta e cioè quando la soglia di tolleranza è molto bassa, allora la probabilità che l'utente reale non venga riconosciuto, è più alta. Tale probabilità scende man mano che la soglia di tolleranza tende ad aumentare.

Quando si progetta un sistema biometrico, bisogna individuare la soglia ottimale (EER) che incrocia le due curve FRR e FAR. In questa logica si fa lavorare per un certo periodo il sistema biometrico facendo variare la soglia di tolleranza e gli errori mappando le due curve, ed infine si raggiunge all'intersezione delle due, che rappresenta un giusto compromesso. Spesso nella progettazione di un sistema biometrico si tende a privilegiare un sistema biometrico che commetta un numero alto di errori FRR a patto di minimizzare gli errori FAR. **Si preferisce avere una soglia di tolleranza molto bassa che porterà ad avere un certo numero di soggetti reali, rifiutati, ma che limita molto l'accesso del numero di impostori.**

VARIABILITÀ DELLE BIOMETRIE:

I dati biometrici a differenza delle tradizionali metodologie di autenticazione non possono essere persi, trasferiti ad altri o dimenticati. L'acquisizione (digitalizzazione dei dati) può determinare una variabilità delle biometrie:

- **Intra_Classe**, cioè la variabilità delle biometrie per **un individuo**.
- **Inter_Classe**, cioè la similarità delle biometrie **tra individui diversi**.

Entrambe le variabilità rappresentano elementi di disturbo per il riconoscimento dell'individuo attraverso tecniche biometriche.

La variabilità **Intra_Classe** può essere resa variabile dal *medio/breve termine*. Alcune biometriche sono condizionate dal trascorrere del tempo.

Il lungo termine è l'età, la posa, le parziali occlusioni, l'espressione e l'illuminazione.

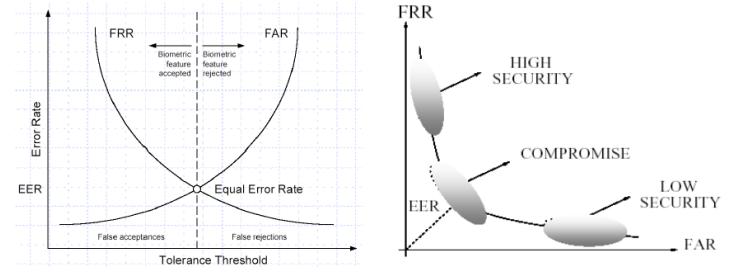
La variabilità **Inter_Classe**, invece, è legata alla presenza di gemelli, sosia, ecc. E quindi la similitudine che ci può essere tra soggetti diversi.

Esempi di variabilità Intra Classe:

- Il volto subisce delle trasformazioni piuttosto limitate in un arco di tempo di due anni. Le impronte digitali invece rimangono intatte.
- Legata anche alla posa, dove un soggetto che cambia la posa può rendere più difficile la propria autenticazione.
- La variazione di espressione può determinare trasformazioni del volto che superano la tolleranza imposta su un sistema di riconoscimento biometrico, ma anche parziali occlusioni (occhiali).
- Con il variare dell'illuminazione il processo di identificazione potrebbe diventare molto ostico, pur essendo il soggetto in una posa corretta.

Esempi di variabilità Inter Classe:

- Legata ai rapporti di parentela (gemelli, ecc).



5. MODELLI DI COLORE

L'occhio umano possiede uno **spettro visibile**, per avere un'idea della gamma di colori che l'occhio umano possa percepire, si fa riferimento alla **luce bianca che attraversa un prisma ottico**.



La luce visibile in realtà non è nient'altro che una serie di **onde elettromagnetiche** nell'intervallo [380nm, 780nm].

- **Ultravioletto** < 380 nm (<10nm Raggi X, Raggi Gamma, Raggi Cosmici)
- **Infrarosso** > 780 nm (>1500nm onde radio)

La percezione del colore avviene attraverso la combinazione di tre stimoli primari (**tricromia**): a livello della retina, grazie a recettori, viene filtrato il segnale e diviso in tre colori: blu, rosso e verde (verde-giallo).



Il colore di un oggetto è percepito come **luce riflessa** dall'oggetto stesso. Un oggetto che **riflette** luce in modo omogeneo sull'intero spettro visibile è percepito di colore **bianco**. Un oggetto che **assorbe** luce in modo omogeneo sull'intero spettro visibile è percepito di colore **nero**. L'occhio è più sensibile al verde, esso riflette luce la cui lunghezza d'onda ricade nell'intervallo [500 nm, 570 nm] e assorbe quasi totalmente le altre lunghezze d'onda [380 nm, 499 nm] U [571 nm, 780]. Il nostro occhio, a differenza dell'orecchio, non è in grado di effettuare un'analisi spettrale, ovvero riconoscere le esatte percentuali di colore, ma riporta una sensazione risultante dalla **combinazione** di tutte le lunghezze d'onda visibili.

Il nostro occhio ha tre famiglie di coni, quelli sensibili al rosso, un'altra sensibile al verde ed un'ultima al blu.

PERCEZIONE E COLORI DEL CIE (COMMISSION INTERNATIONALE DE L'ECLAIRAGE):

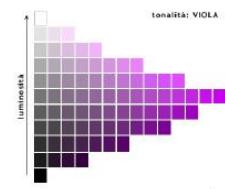
Questa organizzazione si preoccupò di classificare i colori, come colori primari:

Red (700 nm) - Green (546.1 nm) - Blue (435.8 nm)

Ne fu classificato anche un quarto che è una combinazione lineare dei 3 colori primari, detto **luminanza dello stimolo**:

$$Y = 0.299\text{Red} + 0.587\text{Green} + 0.114\text{Blue}$$

Se si considera la tonalità viola, questa la si può misurare sia in termini di saturazione che di luminosità.



La **luminosità** è importante siccome fa cambiare la percezione di un colore, mentre la **saturazione** è la purezza del colore, cioè quando quel colore non è contaminato dal grigio.

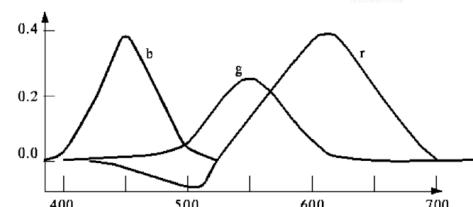
I coni (visione fotopica) sono divisi in tre gruppi:

L (long) sensibili al rosso - M (middle) sensibili al verde - S (short) sensibili al blu.

I bastoncelli sono sensibili nello stesso modo (invarianti) alle varie frequenze dello spettro luminoso: producono una visione (scotopica) basata solo sulle differenze di luminosità.

Percentuale (Y) di intensità dei tre primari (RGB) per comporre tutti i colori dello spettro visibile (X).

I coefficienti negativi indicano che alcuni colori non possono essere rappresentati dalla somma dei tre primari, quindi non riusciamo a vedere tali colori.



Formalmente, nei nostri occhi quando una sorgente luminosa o radiazione (E) passa attraverso la pupilla e colpisce la retina, in base alla lunghezza d'onda (λ), i coni si incominciano a muovere seguendo una legge fisica. Quindi, ogni gruppo di coni somma l'energia luminosa visibile in base alla sua sensibilità e trasmette al sistema visivo la somma dei contributi di ogni lunghezza d'onda pesati dalla sua funzione di sensibilità spettrale (S_R , S_G , S_B).

$$R = \int_{380}^{780} E(\lambda) S_R(\lambda) d\lambda, G = \int_{380}^{780} E(\lambda) S_G(\lambda) d\lambda, B = \int_{380}^{780} E(\lambda) S_B(\lambda) d\lambda$$

MODELLI DI COLORE:

Classi di Immagini (digitali) a Colori:

1. **True colors** (Colori Veri) è ottenuta mediante composizione (sottrattiva o additiva di tre componenti - HSB, RGB, CYM, YIQ). Ogni componente è quantizzata con un numero definito di bit.

2. **Pseudo-Colors** (Colori Falsi) è ottenuta assegnando ad ogni intervallo di colori veri un colore medio.

Il **modello di colore RGB** è utilizzato per realizzare dispositivi di proiezione quali monitor, TV, nell'elaborazione di immagini e immagini satellitari. Esso è un modello **Additivo**, ovvero si addiziona luce (RGB) al nero.

Il **modello di colore CYM** è utilizzato per realizzare dispositivi di stampa. Esso è un modello **Sottrattivo**, ovvero si sottrae luce (CYM) al bianco.

Il **modello di colore HSB** (HSV) è utilizzato nell'Elaborazione di Immagini. Esso è un modello **Combinazione** di Hue (Tonalità), Saturazione (Saturation), Luminosità (Brightness).

Il **modello di colore YIQ** (YUV) è utilizzato nelle trasmissioni TV e nell'elaborazione di immagini (compressione video). **L'occhio umano è più sensibili alla variazione di luminosità che alle variazioni di tono e saturazione.** Sfrutta la maggiore sensibilità dell'occhio umano alla **luminanza** (livelli di grigio).

CONVERSIONE TRA I MODELLI DI COLORE:

Il passaggio tra un modello di colore all'altro avviene attraverso una trasformazione lineare:

Da CYM a RGB è uno l'inverso dell'altro (il valore 1 rappresenta il valore massimo della quantizzazione utilizzato), mentre da YIQ a RGB la matrice ha valori diversificati.

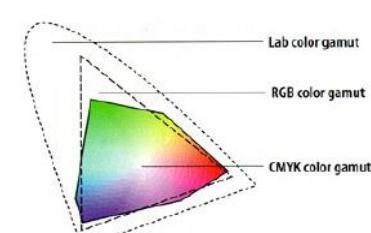
$$\begin{bmatrix} C \\ Y \\ M \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

RELAZIONE TRA MODELLI DI COLORI:

Il **gamut** è l'insieme dei colori che possono essere realizzati dalla combinazione di tre primari. Il modello **LAB** (L = Luminosità, A = asse verde-rosso, B = asse blu-giallo) copre tutti i colori nello spettro visibile.

Il gamut **RGB** è minore del **LAB**, quindi alcuni colori (giallo puro, ciano puro) non possono essere visualizzati sul monitor. Il gamut **CMYK** è il più piccolo (ma non è un semplice sottoinsieme del gamut RGB).

Infatti, il modello CMY e RGB non sono sovrapposti, quindi non rappresentano gli stessi colori. Ad esempio, se si ha una immagine RGB (sul monitor) e si stampa tale immagine, la si converte da RGB a CMY. Alcuni colori possono essere approssimati, perché certi colori non possono essere mai resi o l'occhio umano non riesce a percepire, ma i sistemi informatici riescono a leggerli.



PALETTE OTTIMIZZATA:

È possibile ottimizzare la riduzione dei colori scegliendo una palette che rappresenti al meglio lo spazio dei colori dell'immagine. Una possibile tecnica per ottenere una palette ottimizzata è quella proposta da Paul Heckbert nel 1980, nota come **Median Cut Algorithm**:

1. Si calcola l'**istogramma tridimensionale** dei colori contenuti nell'immagine;
2. Si trova il minimo parallelepipedo (box) orientato come gli assi che contenga tutti i colori.
3. Si sceglie la diagonale del box e si taglia nel punto che lascia lo stesso numero di elementi nelle due metà (taglio mediano).
4. Si ripete il taglio del box con più elementi (maggior numero di colori) fino a quando non si sono ottenuti tanti box quanti i colori richiesti. I colori rappresentativi saranno dati dalla media dei colori inclusi nel box.

6. PANORAMICA SULLE PRINCIPALI BIOMETRICHE

Con la biometria si è cercato di evidenziare i vantaggi di un approccio biometrico per l'identificazione di un individuo, ma esistono limiti e vulnerabilità.

Il primo elemento che rende vulnerabile un sistema biometrico è il **Digital Spoofing**, che avviene **dopo** che il dato biometrico è stato digitalizzato:

- **Attacchi replay**: un hacker ruba l'immagine digitalizzata e se ne serve in un'altra occasione;
- **Attacchi di manipolazione** del valore di soglia tipico di ciascun sistema (obiettivo: aumentare il FAR);
- **Inserimento nel sistema di un "cavalo di Troia"** per fornire dati erronei al programma di estrazione dei parametri biometrici dall'immagine scansionata;
- **Alterazione del risultato finale** del processo biometrico:
l'inserimento e l'analisi dei dati sono corretti, ma il risultato generato dal sistema è alterato.

Il secondo elemento è il **Physical Spoofing**, che è una caratteristica specifica della biometria ed è un tentativo di spoofing che avviene **prima** della digitalizzazione (biometrica sostituita o corrotta):

- **Simulazione** fisica della biometrica

NOTA: Lo **spoofing** è un tipo di attacco informatico che impiega in varie maniere la falsificazione dell'identità (spoof).

FINGERPRINT (IMPRONTE DIGITALI):

Una delle principali applicazioni dei sistemi biometrici è nel contesto dell'**identificazione forense**.

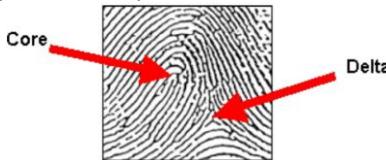
Le impronte digitali è la tecnologia biometrica in uso da più tempo (dalla fine dell'800), ed è nata una vera e propria scienza che oggi viene definita come **Ridgeology**, ovvero lo studio dell'unicità delle strutture delle creste e delle valli presenti sulle impronte per l'identificazione personale.

Un'impronta digitale è composta da una serie di creste e valli sulla superficie del dito. L'unicità di un'impronta digitale può essere determinata dal modello di creste e valli, nonché dai **punti di minuzie** che sono i punti in cui cambia la struttura della cresta.

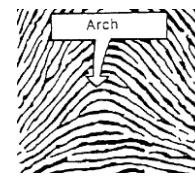
Le **caratteristiche globali** possono essere viste anche ad occhio nudo (spirali, archi e loop), ma non sono sufficienti per il riconoscimento e vengono utilizzate solo per **classificare** (clustering) le impronte (divisione in classi). Gli archi sono più rari, mentre sono più frequenti i loop e le spirali. Queste caratteristiche sono macro-caratteristiche, quindi, possono essere condivise da più di una persona e come tali non consentono la discriminazione della persona stessa. Per individuare le tre caratteristiche bisogna trovare dei **punti focali**:

LOOP: Esistono due punti focali che accertano la sua presenza:

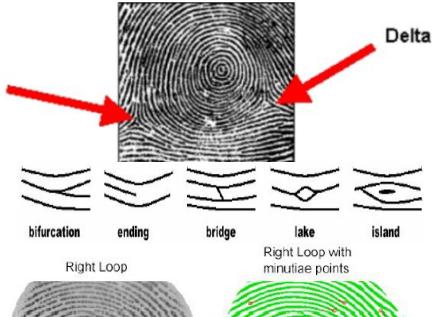
- **Core**, parte terminale del loop (più interna);
- **Delta**, punto in cui è presente una biforcazione.



ARCH: Nell'arco **non** sono presenti **core** e **delta**.



WHORL: Nelle spirali si hanno **due delta**, perché le linee provengono sia da sinistra che da destra per poi biforcarsi.



Le **caratteristiche locali** dette **minuzie (singolarità)** si utilizzano solo per il **riconoscimento** all'interno della classe prodotta dalle caratteristiche globali. Queste caratteristiche sono invece estremamente singolari, anche dal punto di vista dell'appartenenza, nel senso che ogni individuo ne possiede un numero diverso con una distribuzione topologica diversificata all'interno dell'impronta digitale. Le minuzie sono quelle evidenziate in rosso. È possibile trovare due persone con caratteristiche globali praticamente equivalenti ma con caratteristiche locali (minuzie) assolutamente diverse. Le minuzie, quindi, sono l'elemento attraverso il quale si riesce ad effettuare una classificazione delle impronte digitali. Esse vanno individuate nell'impronta digitale, individuando la **tipologia** e la loro **lokazione spaziale**. In generale su ogni impronta digitale si possono rilevare fino a **150 minuzie**. Per dimostrare che l'impronta digitale appartiene ad una determinata persona, delle 150 minuzie potenzialmente presenti sull'impronta digitale, in un contesto legale, ne bastano *poco più di una decina*.

Il livello delle minuzie non è il livello più dettagliato che si può avere in una impronta digitale, infatti per aumentare il livello di sicurezza si vanno a localizzare un certo numero di **pori** per la sudorazione lungo una singola linea crestale.

I pori rappresentano un livello di dettaglio ultrafine, in genere per acquisire tali pori si ha bisogno di uno scanner con una risoluzione più tosto significativa.



FINGERPRINT IDENTIFICATION:

Relativamente alle caratteristiche locali, si è interessati a individuare le minuzie ma anche alle relazioni che intercorrono tra di loro. Per rappresentare queste relazioni si utilizza una **struttura grafo** dove ogni nodo rappresenta il tipo di minuzia, mentre gli archi rappresentano la relazione spaziale.

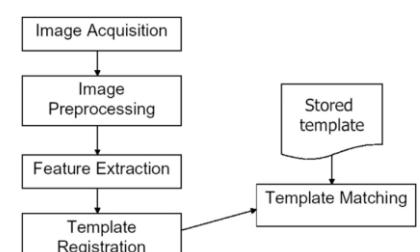
Quando viene effettuato un matching tra impronte si confrontano i grafi generati, questo confronto è un problema **non-polinomiale**. Infatti, per accelerare i tempi, si cercano prima le **caratteristiche globali** per individuare la classe di appartenenza dell'impronta, e poi si individuano le **minuzie**.

Non viene effettuata direttamente la ricerca delle caratteristiche locali perché è un processo oneroso.

I sistemi per la ricerca delle impronte digitali prendono il nome di **AFIS (Automated Fingerprint Identification System)**, ognuno di esso può utilizzare un sistema di identificazione (o algoritmo) diverso.

La struttura di un algoritmo per il riconoscimento di una impronta digitale è:

1. **Image acquisition**, l'impronta digitale viene acquisita e le modalità sono differenti (latente o dal vivo);
2. **Image processing**, l'immagine viene pre-elaborata per ottimizzare l'estrazione delle caratteristiche;
3. **Feature extraction**, vengono estratte le caratteristiche delle impronte;
4. **Template registration**, dalle caratteristiche estratte si crea un template strutturato dalle caratteristiche globali e locali;
5. **Template matching**, viene confrontata l'impronta col template già registrato.



FINGERPRINT - VANTAGGI vs SVANTAGGI:

Per quanto riguarda i vantaggi abbiamo:

- **Universalità**, ovvero quella caratteristica presente in tutti gli individui ad eccezione di persone con impronte danneggiate volontariamente o meno;
- **Unicità**, non esistono impronte uguali se consideriamo le caratteristiche locali;
- **Permanenza**, le impronte persistono per sempre a meno di abrasioni o invecchiamento;
- **Misurabilità**, attraverso le minuzie, è possibile misurare un'impronta;
- **Accettabilità**, volontà o gradimento della persona a farsi riconoscere;
- **Efficacia**, rappresentazione algoritmica dell'unicità;
- **Acquisizione**, può essere attiva se si effettua una misurazione volontariamente, mentre passiva se la si usa in ambito forense.

Mentre gli svantaggi sono:

- **Efficienza**: non viene memorizzata l'immagine dell'impronta, ma un suo template (chiave di ricerca), ad eccezione dell'AFIS che vengono memorizzate le immagini perché devono essere verificati da un perito, e per estrarre le caratteristiche ed effettuare un matching richiede del tempo;
- **Stabilità**: presenza di abrasioni, rughe, sporco, sudore, disidratazione (over 65), inibendo le caratteristiche dell'impronta;
- **Insidia**: relativamente facile da duplicare (Physical spoofing).

DIGITALIZZAZIONE DELLE IMPRUNTE DIGITALI:

Quando una impronta non è latente si utilizzano gli scanner, che possono essere di tipo differente. I parametri che caratterizzano uno scanner sono:

- la **risoluzione**, legata al numero di punti di unità di misura che lo scanner è in grado di rappresentare;
- l'**area di acquisizione**, più è ampia l'aria acquisita più è probabile che l'impronta possa essere acquisita integralmente;
- il **numero di pixel**, legata alla risoluzione;
- l'**accuratezza geometrica**, che insieme alla distorsione geometrica è legato al fatto che avendo posizionato il dito vicino al sensore ci sono delle distorsioni prodotte dalla necessità di mettere a fuoco un oggetto con particolari moto piccoli ad una distanza ristretta dalla focale;
- il **contrasto**, procedura di ottimizzazione dell'immagine da acquisire.

I sensori di impronte digitali possono utilizzare le seguenti tecnologie:

- **FTIR Frustrated Total Internal Reflection (Ottica)**, in grado di rilevare le immagini di creste e vallate con due diversi livelli di messa a fuoco;
- **Capacitiva (dispositivi mobili)**: basata sul rilevamento delle microvariazioni di capacità (accumulo di carica elettrica) originate dalla struttura tridimensionale del polpastrello;
- **Piezoelettrica**, basata sul rilevamento dell'impronta tramite materiali in grado di convertire la variazione di pressione locale in una differenza di potenziale elettrico;
- **Termica**, basata sull'utilizzo di sensori termici in grado di rilevare l'immagine del polpastrello non in lunghezza d'onda visibili ma nella gamma dell'infrarosso.

7. FACE DETECTION E RECOGNITION

Il **volto** è una “biometria al limite”, perché esistono diversi limiti legati alle caratteristiche che vengono utilizzate per il riconoscimento del volto, sono caratteristiche molto mutabili che hanno una trasformazione in un arco temporale medio. Nei sistemi di riconoscimento (videosorveglianza) il volto è la biometria più utilizzata, per una serie di vantaggi come l'*acquisizione passiva*.

La capacità di un individuo di discriminare i volti porta ad una serie di situazioni particolari, come ad esempio la **pareidolia**, che è l'illusione subcosciente che tende a ricondurre a forme note, oggetti o profili (naturali o artificiali), dalla forma casuale.

Relativamente al volto esiste una patologia, ovvero la **prosopagnosia**, che è un deficit percettivo acquisito o congenito del sistema nervoso centrale che impedisce ai soggetti che ne vengono colpiti di riconoscere correttamente i volti delle persone.

Oggi il volto è largamente utilizzato, nei sistemi forensi e non solo, per due motivi principali:

- **Affidabilità**, molto affidabile nella verifica (match 1 a 1) o in generale quando il soggetto è cooperativo o viene acquisito in condizioni ottimali;
- **Accettabilità**, largamente accettato come sistema di riconoscimento.

Esistono altre biometriche, come l'iride e le impronte, che hanno prestazioni in termini di affidabilità più elevate ma risultano più difficili da acquisire. Il volto, tra le biometriche, non è quella che fornisce prestazioni più elevate in termini di riconoscimento ma è quella che ha il compromesso migliore tra prestazioni, affidabilità e accettabilità. Il riconoscimento del volto viene principalmente utilizzato, insieme alle impronte, al contesto forense.

Un problema sostanziale che è legato alla tipologia di biometrica **contactless** sta nel fatto che quando si effettua un'acquisizione di impronta digitale l'intero spazio di acquisizione è legata solo all'impronta, mentre per una immagine contenente volti prima di passare al riconoscimento di un volto è necessario un pre-processing per andarlo ad individuare (**detection** del volto).

La verifica e il riconoscimento sono le due tipologie di applicazione in cui il volto può essere utilizzato.

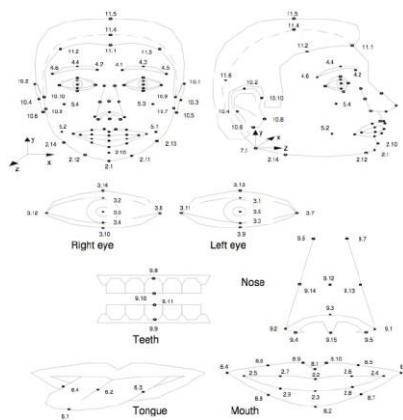
- **Verifica**: confronto Uno a Uno. Conferma l'identità di un individuo dichiarata mediante carta d'identità, codice utente, ecc...
- **Riconoscimento**: confronto Uno a Molti. Stabilisce l'identità di un soggetto a partire da un insieme di persone registrate.

DIGITALIZZAZIONE DEL VOLTO:

Per quanto riguarda la struttura del volto questa è **tridimensionale** ma quando si *digitalizza il volto* diventa **bidimensionale**, perdendo ad esempio la misure del naso o quanto sono incavati gli occhi rispetto alle sopraccigli. Per non perdere queste informazioni si dovrebbe avere un doppio sistema di telecamere che simula la **coppia stereo** dell'occhio (guardando con un occhio solo si perde la terza dimensione) oppure esistono le depth camera che sono in grado di creare mappe di profondità dando una informazione tridimensionale.

Un sistema automatico avrebbe maggiore facilità nel riconoscere una persona se si aggiungessero una serie di informazioni legate alla morfologia tridimensionale del volto.

Queste informazioni sono molto importanti per valutare empiricamente i particolari del viso. Il volto ha una struttura molto complesso, ma vengono utilizzate solo un sottoinsieme di queste informazioni, ovvero quelle più evidenti durante l'acquisizione bidimensionale, questo è un fatto importante che va considerato quando si evidenziano gli errori che commettono i sistemi di **face recognition**.



PROBLEMI COL VOLTO:

I problemi legati alla biometria del volto sono diversi, come ad esempio l'eccessiva variabilità tra le persone e la similarità interpersonale (sosia o chirurgia plastica). Un altro aspetto fortemente perturbativo dei sistemi di Face Recognition sono le perturbazioni che possono essere indotte dall'acquisizione, il volto è una delle poche biometriche che può essere variata dall'individuo a differenza di altre, infatti si possono produrre delle variazioni volontarie sul volto (espressioni o smorfie). Quando a queste variazioni si aggiungono ulteriori fattori perturbativi come la variazione di posa e illuminazione, si creano le **PIE variations (Pose, Illumination, Expression)**. Esse sono in realtà le variazioni che rendono più difficile il processo di acquisizione. Questi elementi perturbativi sono elementi che intervengono quando l'utente non è collaborativo.

L'altro fattore che non è indotto dalla persona e dalle condizioni ambientali è il cosiddetto **Ageing (A-PIE, PIE variations + Ageing)**, ovvero la variazione dell'età. Esso è un aspetto fortemente perturbatore nel processo di Face Recognition. Le caratteristiche somatiche tendono a modificarsi in intervalli di tempo molto piccoli, e tendono progressivamente a determinare effetti che col tempo possono rendere inefficaci i processi di riconoscimento.

Quando si sviluppa una tecnica di face Recognition, essa la si testa su database comuni per comprendere in modo comparativo, se l'idea algoritmica prodotta risulta essere migliore rispetto ad altre. Negli anni sono stati prodotti diversi database e quando la comunità scientifica produce un nuovo algoritmo, tende ad utilizzare uno più database per avere una misura comparativa con quello che è stato prodotto in precedenza.

STRUTTURA DI UN RICONOSCITORE FACCIALE:

Il primo passo è legato ad individuare all'interno di un video/immagine la presenza di un volto. Una volta che il volto è stato individuato con qualche tecnica, allora è possibile darlo in input al sistema di classificazione che deve estrarre le caratteristiche principali. Il primo problema da risolvere, è quello di costruire un **face detector efficace**, ovvero come capire se in un'immagine vi è un volto.

FACE LOCALIZATION:

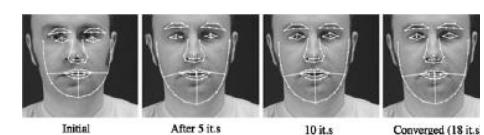
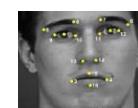
La localizzazione di volti è un problema importante perché, nel momento in cui questa operazione viene affidata ad una procedura automatica ci possono essere degli errori di rilevamento (volto presente e non individuato oppure volto individuato ma non presente).

Le tecniche per effettuare operazioni di Face Detection sono diverse, come le **face detection features-based**, esse sono tecniche che sfruttano alcune proprietà di basso livello del volto per *individuarlo* all'interno di una immagine.

Le più elementari ma *estremamente performanti* sono quelle basate su caratteristiche di basso livello che possono riguardare alcune **caratteristiche dei pixel** (luminosità o geometria del volto).

Infatti, tutti i punti che individuano occhi, narici, sopracciglia ecc... creano una costellazione di informazioni che possono essere legate da una serie di **relazioni geometriche**.

Oppure si può immaginare di avere un **template di un volto** rappresentabile con un modello matematico.



Poi ci sono degli approcci che piuttosto che basarsi sulle caratteristiche elementari del volto sfruttano le immagini nella loro totalità, le cosiddette tecniche **imaged-based**. Queste tecniche si basano sull'addestramento di modelli computazionali, in grado di riconoscere determinati oggetti sulla base di un pre-addestramento effettuato. Si costruisce un dataset di addestramento, dove all'interno ci sono sia immagini contenenti volti (volti campione) e sia immagini non contenenti volti, e si allena il sistema a riconoscere soltanto i volti.

EYE / MOUTH LOCALIZATION:

Una delle tecniche basate sulle **caratteristiche di basso livello** sfrutta la relazione tra il modello RGB e YCC. Quest'ultimo è quello più utilizzato nell'image processing, è un modello nel quale è possibile separare un componente dell'immagine, rispetto a delle altre, concentrando in questa componente (Y) la maggior parte dell'informazione significativa. La transizione tra i due modelli avviene attraverso una trasformazione lineare che mappa il modello RGB nel modello YCC. Nel caso del modello YCC l'informazione che interessa non è contenuta nella componente di **luminanza** Y, ma piuttosto nelle altre due componenti Cr Cb (**crominanza**).

Nella componente di crominanza sono particolarmente evidente alcune caratteristiche del volto, in particolare occhi e bocca, presa questa componente di crominanza si applicano una serie di operatori (equalizzazione e AND) che tendono a massimizzare gli aspetti che interessano occhi e bocca. La loro individuazione permette di avere una certa probabilità che l'immagine in considerazione contenga un volto. Tale tecnica produce molti **falsi positivi** (volti dove non ci sono).

FACE CONTOUR:

Un approccio che rientra più nell'ambito **geometrico** prevede l'individuazione della bocca e degli occhi ma vincola essi ad una certa relazione geometrica. Ovvero deve crearsi un triangolo che gode di alcune proprietà specifiche. Tale triangolo viene anche legato ad esser circoscritto da una circonferenza che rappresenta il volto vero e proprio. Si passa ad un approccio di natura geometrica in quanto ci si basa non solo sulla presenza di occhi e bocca ma anche sulle caratteristiche geometriche che vengono calcolate sulla base di una serie di operatori.

FACE DETECTION – VIOLA-JONES:

Paul Viola e Michael Jones hanno proposto uno degli approcci di maggior successo (finora) alla localizzazione degli oggetti (incluso in OpenCV). L'algoritmo è **image-based** e può essere applicato al rilevamento del volto (ma anche al rilevamento di occhi e bocca in una strategia gerarchica). Esso richiede di **creare un classificatore** inizialmente addestrato utilizzando più istanze della classe da identificare (esempi positivi), e diverse istanze di immagini che non contengono alcun oggetto della classe ma che possono causare un errore (esempi negativi). Il Training è progettato per **estrarre diverse funzionalità** dagli esempi e per selezionare quelle più discriminanti. Il modello statistico che viene costruito in modo incrementale contiene tali informazioni. **Misses** (un oggetto presente non viene rilevato) o **False Alarms** (un oggetto viene rilevato ma non è presente) possono essere ridotti riqualificando l'aggiunta di nuovi esempi adatti (positivi o negativi).

L'algoritmo di Viola-Jones è un algoritmo **composito** in cui ci sono alcuni aspetti che di fatto mettono insieme un po' tutto quello che è stato analizzato in precedenza nell'ambito dell'elaborazione delle immagini. È un algoritmo molto veloce in quanto vengono posti in modo preliminare al processo di riconoscimento vero e proprio, utilizzando dei classificatori, dei filtri e attraverso la combinazione di essi, mette a punto una strategia gerarchica affinché riesca ad individuare se in un'immagine è presente un volto.

L'algoritmo di Viola-Jones sfrutta le cosiddette **caratteristiche di Haar**, cioè delle caratteristiche tipiche del volto, condivise da tutti i volti umani. Queste caratteristiche presenti nel volto vengono rappresentate da dei **filtri** che vengono **sovraposti** all'immagine. Tali filtri cercano di verificare la presenza di alcuni aspetti tipici del volto umano, come ad esempio la regione perioculare (occhi) è generalmente più scura della zona sottostante (zigomi) così come il ponte del naso è normalmente più luminoso degli occhi.

Tendenzialmente si possono utilizzare filtri molto simili a questi per cercare di capire se all'interno di una immagine c'è un'alternanza di regione scura/chiara, perché tale alternanza potrebbe essere sintomo di presenza di occhi e zigomi. Così come un'alternanza nero/bianco/nero potrebbe essere funzionale all'aver individuato occhio/ponte del naso/occhio.

È chiaro che queste sono condizioni necessarie ma non sufficienti per individuare un volto, ed è chiaro *che di questi filtri se ne dovranno considerare di diversi dimensioni* da far passare sul volto al fine di poter avere una certezza che quel che viene isolato è effettivamente un volto.

Una volta individuato il filtro da far scorrere sull'immagine lo si sovrappone su di essa e si verifica la variazione fra zona chiara e zona scura: si moltiplicano per 1 i pixel dell'area bianca e per 0 i pixel dell'area nera e si effettua una differenza, che poi viene controllata se essa **superà una certa soglia**. Se si supera una certa soglia allora vuol dire che quel filtro ha una forte probabilità di aver localizzato una regione dell'immagine contenente un volto.

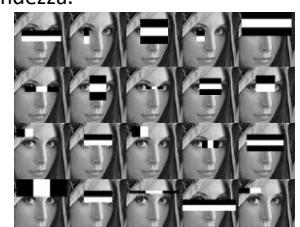
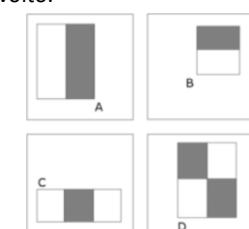
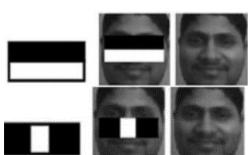
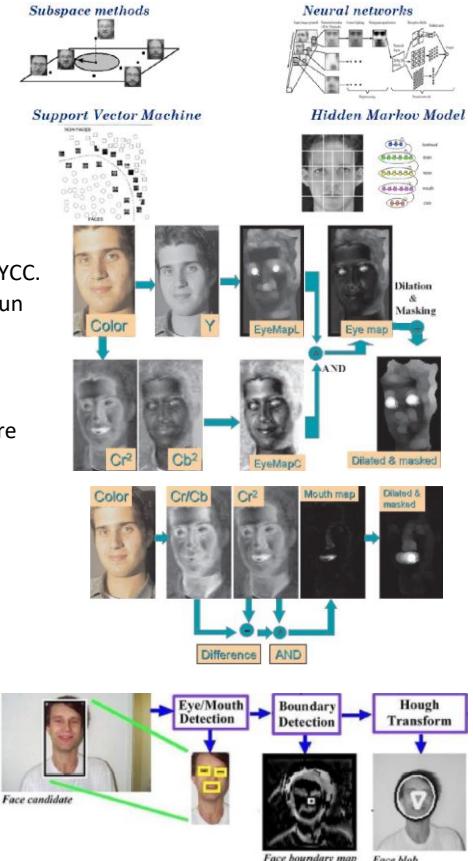
Il valore di una determinata funzione è sempre semplicemente la somma dei pixel all'interno di rettangoli chiari sottratti dalla somma dei pixel all'interno dei rettangoli ombreggiati. Caratteristiche rettangolo:

$$\text{Valore} = \Sigma (\text{pixel nell'area nera}) - \Sigma (\text{pixel nell'area bianca})$$

Esempio Viola-Jones:

Una serie di filtri vengono applicati sull'immagine: diversi filtri dove ognuno dovrebbe individuare una parte significativa dell'immagine. Quando viene individuata un'area potenzialmente assimilabile ad un'immagine il quadro si blocca in quanto si è in presenza del primo frame utilizzabile per classificare l'immagine come volto. I filtri possono avere diverse dimensioni e le finestre possono essere di diversa grandezza.

Come si vede in figura, vengono mostrate alcune delle caratteristiche che possono essere utilizzate da Viola-Jones. Uno dei problemi di tale algoritmo è che dovrebbe unire efficacia ad efficienza, quindi bravo nell'evidenziare i potenziali volti in un tempo piuttosto rapido. Il problema è che ogni volta viene applicata una mascherina come in figura sull'immagine, devono essere effettuate delle semplici **operazioni** (moltiplicazioni e sottrazioni) **ripetute un gran numero di volte** su tutta quanta l'immagine, in quanto tali operazioni vengono effettuate per ognuno dei filtri con dimensione anche variabile dei filtri stessi. **Per ridurre il costo computazionale l'immagine viene trasformata nella cosiddetta Integral Image.**



ALGORITMO VIOLA-JONES:

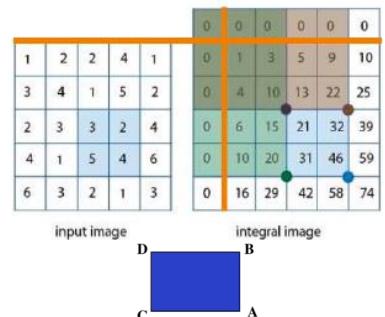
L'algoritmo utilizza delle **Haar features** in combinazione con una nuova rappresentazione dell'immagine detta **Integral Image**. Le features hanno basso costo computazionale e la nuova struttura dati permette di effettuare l'analisi in tempo costante indipendentemente dalla dimensione delle regioni analizzate. Viene introdotto un metodo di selezione di feature di Haar attraverso l'algoritmo AdaBoost di Freud Shapire. Questa strategia permette di eliminare in addestramento la maggior parte delle feature di scarsa capacità discriminante e selezionare solo quelle più efficaci per il problema.

INTEGRAL IMAGES:

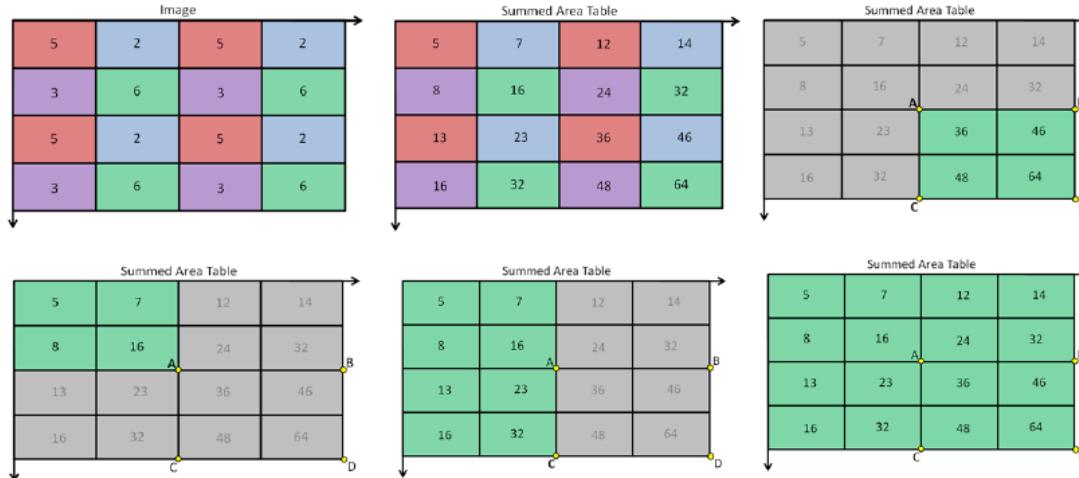
L'immagine integrale non è nient'altro che una rappresentazione/trasformazione più efficiente dell'immagine per realizzare queste moltiplicazioni e sottrazioni. Come in figura, l'immagine integrale viene costruita sommando i pixel in maniera progressiva. Quindi ogni pixel rappresenta la somma dei pixel precedenti in maniera crescente partendo *da sinistra verso destra e dall'alto verso il basso*. La rappresentazione dell'immagine in termini di immagine integrale permette di realizzare la somma dei pixel in una **qualsiasi area** (rettangolo o quadrato di riferimento) attraverso un numero più efficiente di operazioni, ovvero **soltamente tre operazioni**. Indipendentemente dalla dimensione del filtro si potrà sempre realizzare la somma dei pixel all'interno di un'immagine in termini di: A - B - C + D facendo fondamentalmente solo tre operazioni elementari, questo risulta essere un grande vantaggio in quanto applicando un filtro di HAAR sull'immagine, si può avere in tempo costante (le sole tre operazioni) indipendentemente dal numero di pixel che sono coperti, la somma dei pixel.

Poiché fondamentalmente nei filtri di HAAR si deve effettuare la somma dei pixel per poi fare la differenza, la riuscita di queste operazioni velocemente è un grande vantaggio.

$$II(x_0, y_0) = \sum_{x=0}^{x_0} \sum_{y=0}^{y_0} I(x, y);$$

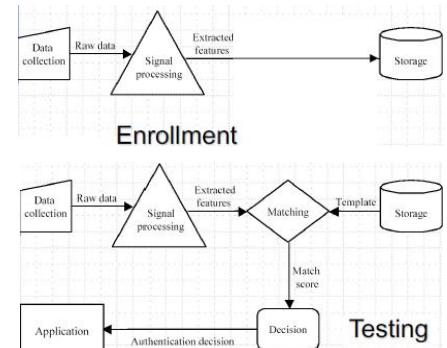


Esempio Integral Images:



FACE RECOGNITION:

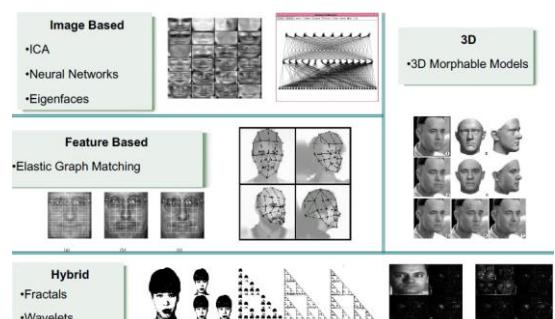
Una volta effettuata l'operazione di **detection**, ovvero individuato il volto all'interno di un'immagine, successivamente si procede all'estrazione delle caratteristiche del volto che permettono di effettuarne il riconoscimento. Per poter fare il **riconoscimento** del volto bisogna avere un **template**, ovvero un modello corrispondente affinché possa essere confrontato e verificato. In precedenza, si è accennato che un sistema di riconoscimento biometrico prevede due fasi: **Enrollment** in cui si crea il template, dove il dato viene acquisito attraverso un sensore e le informazioni estratte con l'image processing vengono catalogate sotto forma di template in un database. Questo processo può essere effettuato singolarmente o in blocco (**batch Enrollment**). L'altra fase, quella di **Testing** prevede l'acquisizione nuovamente dei dati attraverso la stessa tipologia di sensore, l'estrazione delle caratteristiche e il recupero del template dal database per il confronto con i dati estratti. Attraverso una soglia di tolleranza si verifica se il **matching** produce un risultato positivo o negativo.



SISTEMI ESISTENTI IN LETTERATURA:

In letteratura esistono svariati approcci di tecniche di Face Recognition. Sostanzialmente si possono individuare quattro famiglie di **algoritmi** di **Face Recognition**:

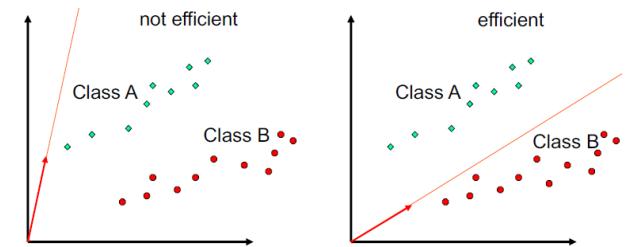
1. **Image Based**, l'immagine, nella sua interezza, viene considerata come base di partenza dal quale estrarre delle caratteristiche facciali;
2. **Feature Based**, sull'immagine si individuano alcuni punti cardine (hotspot) su cui è possibile costruire dei grafici;
3. **3D**, si basa su modelli tridimensionali, acquisizione con più telecamere o depth camera;
4. **Hybrid**, annovera tutti i restanti algoritmi che non rientrano nelle precedenti famiglie.



ICA (INDEPENDENT COMPONENT ANALYSIS) E PCA (PRINCIPAL COMPONENT ANALYSIS):

Tra le tecniche più utilizzate soprattutto in passato per il Face Recognition c'è la **PCA** che rientra nella categoria delle **Image Based**. Piuttosto che considerare i pixel, essi vengono associati a delle variabili geometriche, ovvero vettori (le direzioni) in cui è massima la variazione (varianza) tra i vettori stessi. L'**obiettivo** della PCA è di individuare dei vettori che siano efficienti per rappresentare determinati dati. L'**ICA** (**più precisa**) prende in considerazione tutti i vettori che variano, mentre la **PCA** (**più veloce**) prende solo i vettori principali, ovvero quelli più variabili in termini di covarianza.

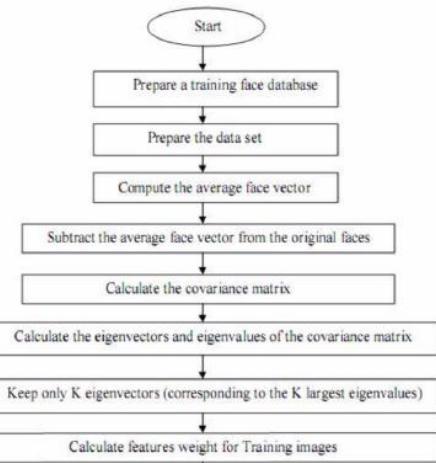
Si veda in figura, si supponga di avere due classi di dati A e B, nella prima rappresentazione non è possibile affermare che il vettore possiede una direzione rappresentativa della classe A o della classe B, infatti la rappresentazione non indica il comportamento delle due classi. Esse pur avendo dati distinti, possiedono delle caratteristiche in comune, ovvero una distribuzione delle informazioni rispetto al piano cartesiano piuttosto simile. Nella seconda rappresentazione invece, il vettore è rappresentativo delle due classi, infatti, il vettore e le due classi condividono la direzione e quindi il coefficiente angolare della retta associata.



Si assume di avere delle **immagini quadrate di dimensione N** , in cui il numero di pixel risulta essere $N \times N$. Con M il **numero di immagini del database** e P il **numero di persone coinvolte all'interno del database**, quindi con M diverso da P (per ogni persona si possono avere più immagini nel database). L'algoritmo della PCA non fa nient'altro che cercare di individuare all'interno di un insieme di immagini, le **immagini più significative**, quindi tentare di trovare un numero di immagini molto più piccolo di M tale che queste immagini siano **rappresentative** di tutto il dataset. Quindi trovare una strategia che in presenza di un dataset molto corposo, possa dare in output un **insieme molto più piccolo** di immagini significative. Tali immagini significative possono rappresentare tutto il dataset. Riassumendo si tenta di ridurre il problema di rappresentazione ad un problema più piccolo trasformando la dimensionalità del problema. L'idea è quella di rappresentare **ogni volto** attraverso la **combinazione lineare** dei volti più significativi, i cosiddetti **Eigenfaces**. Quindi ogni volto significativo contribuirà alla rappresentazione di un volto all'interno dell'immagine.

PCA – METODO:

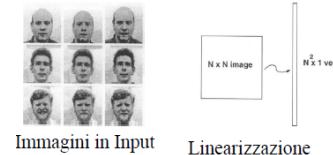
- Training set**, costituito da un insieme di immagini delle stesse dimensioni e normalizzate in maniera da evidenziare caratteristiche del volto, quali occhi, naso e bocca.
- Proiezione in un sottospazio**, nel quale vengono messe in risalto le caratteristiche salienti di un volto. Tale riduzione dimensionale, infatti, permette di escludere l'informazione che non viene considerata rilevante e precisamente decomponete la struttura di un volto in una combinazione di componenti ortogonali, scorrelate tra loro, dette **eigenface**.
- Ogni immagine di volto**, può successivamente essere rappresentata come una somma pesata (vettore delle feature) di queste *eigenfaces*, raccolte in un vettore monodimensionale. Il confronto di un'immagine di volto con le altre presenti nel training set viene effettuato semplicemente valutando la distanza tra questi vettori di caratteristiche locali.
- Matrice di Covarianza**: Matrice Simmetrica che rappresenta la variazione di ogni variabile rispetto alle altre (inclusa sé stessa).
 - Se valori grandi" (rispetto il valore atteso) di X si accostino a valori grandi di Y, oppure "valori piccoli" (rispetto il valore atteso) di X con valori piccoli di Y, la covarianza è ≥ 0 . In questo caso si dice che X e Y sono **correlate positivamente**.
 - Se invece valori piccoli di X tendono ad accoppiarsi con valori grandi di Y o viceversa, la covarianza sarà ≤ 0 e le variabili aleatorie in questione si diranno **correlate negativamente**.



EIGENFACES – SISTEMI LINEARI:

Si prendono in input le immagini e per ognuna di esse si effettua una **linearizzazione**, ovvero ogni immagine viene rappresentata come vettore, si calcola la **media** su tutti quanti i vettori e tale media viene **sottratta** da ciascun vettore. Quest'ultima operazione fa in modo che venga rimossa la parte in **comune** tra tutti i volti, ottenendo quindi le immagini che andranno a costituire la base per la matrice di covarianza, si scelgono le immagini **meno correlate** tra di loro (quelle più vicine ad essere linearmente indipendenti). Una volta individuato l'insieme dei vetri (immagini) meno correlate, è possibile rappresentare tutta la base di dati attraverso una **combinazione lineare** degli *Eigenfaces*.

Quindi la scelta delle immagini più indipendenti e più significative permette la generazione di tutta quanta la base di dati.



$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i, \quad \Phi_i = \Gamma_i - \Psi$$

Media su tutte i volti A ciascun vettore togliamo la media



Risolvendo un particolare sistema lineare. I coefficienti utilizzati per la rappresentazione costituiscono il vettore di caratteristiche

$$\text{Ciascun volto viene rappresentato come una combinazione dei volti di riferimento. I coefficienti utilizzati per la rappresentazione costituiscono il vettore di caratteristiche}$$

Media su tutte i volti A ciascun vettore togliamo la media

EIGENFACES – ALGORITMO:

PASSO 1: Ogni volto viene trasformato in un vettore, nello spazio N^2 . In questo spazio poi bisogna trovare la base, quindi il numero minimo di vettori linearmente indipendenti in grado di rappresentare tutta quanta la base attraverso la combinazione lineare.

PASSO 2: Viene calcolata la media, che ci permette di individuare quale è la parte comune a tutti i vettori e viene sottratta poi da ogni vettore.

PASSO 3: il processo più importante è quando si costruisce la matrice di covarianza, che ci permette di individuare quali di questi vettori sono candidati a diventare auto-vettori (vettori rappresentativi di tutta la base di riferimento).

Covarianza positiva: indica che due variabili tendono a muoversi nella stessa direzione, e quindi sovrapponibili (hanno le stesse caratteristiche).

Covarianza negativa: rivela che due variabili tendono a muoversi in direzioni inverse (candidabili a rappresentare la base, quindi linearmente indipendente).

$$\vec{m} = \frac{1}{M} \begin{pmatrix} a_1 + b_1 + \dots + h_1 \\ a_2 + b_2 + \dots + h_2 \\ \vdots \\ a_{N^2} + b_{N^2} + \dots + h_{N^2} \end{pmatrix}, \quad \text{where } M = 8$$

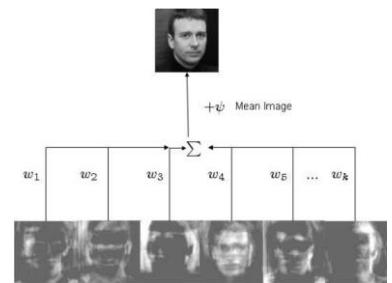
$$A = [\vec{a}_m \vec{b}_m \vec{c}_m \vec{d}_m \vec{e}_m \vec{f}_m \vec{g}_m \vec{h}_m]$$

$$\text{Cov} = AA^T$$

PCA ED EIGENFACES:

La rappresentazione di ogni volto all'interno del database avviene attraverso la combinazione lineare dei vettori linearmente indipendenti o che quelli che si sono classificati come potenzialmente linearmente indipendenti.

L'obiettivo è quello di avere una serie di auto-vettori, le immagini in basso rappresentano le immagini alle quali sono state sottratte le medie che in combinazione lineare danno origine al volto di riferimento, ed infine aggiungere la media tolta in precedenza.



Riassumendo, il principio alla base della PCA è la ricerca all'interno della base di dati delle immagini meno correlate aspettandosi che esse siano dal punto di vista numerico molto inferiore alla dimensione dell'intero database. Quindi, se il database possiede M immagini, si cercano k immagini con k molto piccolo, tale da poter rappresentare tutte le immagini del database attraverso la combinazione lineare delle k scelte. Esse vengono scelte grazie alla matrice di covarianza che permette di stabilire quali immagini risultano essere meno correlate tra esse, meno le immagini sono correlate più sono tra di loro indipendenti. Le immagini vengono assimilate da dei vettori *n-dimensional*. La matrice di covarianza permette di stabilire quante sono le immagini. La scelta delle immagini dipende anche dall'**accuratezza** che si vuol ottenere. Più è **piccola** la dimensione del set di Eigenfaces scelto, **meno** è l'accuratezza ottenuta nella rappresentazione delle immagini (maggior errore).

VANTAGGI

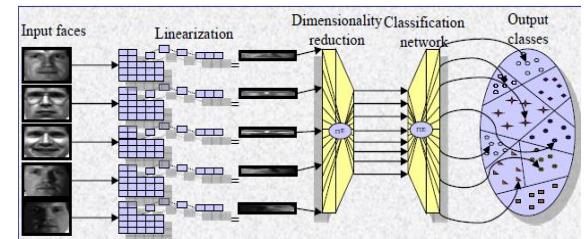
- La fase di identificazione è veloce;
- Funziona bene quando la **base di dati non cambia** o quando è abbastanza stabile;
- Se si conservano gli auto-vettori è possibile ricostruire l'informazione originaria.

SVANTAGGI

- La fase di training è lenta.
- Se si aggiunge un numero consistente di nuovi soggetti è **necessario il retraining** del sistema.
- Elevata sensibilità a variazioni di illuminazione, posa, occlusioni, ...

RETI NEURALI:

Una rete neurale simula il funzionamento dei neuroni nel cervello. Ciascun neurone è rappresentato da una funzione matematica basata sul calcolo delle probabilità. Per il riconoscimento dei volti, l'ottimo sarebbe utilizzare un neurone per ciascun pixel. Questo approccio richiede troppi neuroni. La soluzione consiste nell'utilizzare una rete di neuroni per riassumere l'immagine in un vettore più piccolo. Mentre una seconda rete effettua il riconoscimento vero e proprio.



VANTAGGI

- Riducono l'ambiguità fra soggetti appartenenti a classi simili;
- Con opportuni accorgimenti sono robuste alle occlusioni.

SVANTAGGI

- Richiedono più di una immagine per il training;
- Alcune reti sono soggette a problemi di:
 - **Overfitting**: la rete ha la stessa dimensione dell'input;
 - **Overtraining**: se le immagini sono grandi il metodo converge lentamente;
 - **Database size**: quando il numero di soggetti cresce diventano inefficienti.

SISTEMI BASATI SU GRAFI:

Attraverso filtri e funzioni di localizzazione vengono localizzati sul volto un insieme di punti di riferimento. Questi punti vengono collegati da archi pesati e si ottiene un grafo. Ad ogni volto è associato un grafo, per cui confrontare due volti significa confrontare due grafi.



VANTAGGI

- Sono robusti rispetto alle variazioni di posa;
- Sono robusti rispetto alle variazioni di illuminazione
- Non richiedono il retraining del sistema

SVANTAGGI

- Il processo di testing è molto lento perché richiede il confronto fra grafi (NP-Hard).

TERMOGRAMMA:

L'immagine del volto viene acquisita mediante un sensore termico. Il sensore rileva le variazioni di temperatura dell'epidermide del volto. L'immagine viene segmentata e indicizzata.



VANTAGGI

- Sono robusti rispetto alle variazioni di illuminazione;
- Sono robusti rispetto alle variazioni di tempo;
- Sono efficienti anche in caso di ambienti esterni.

SVANTAGGI

- Richiedono dispositivi di acquisizione costosi;
- I dispositivi di acquisizione sono troppo sensibili ai movimenti del soggetto e offrono limitate risoluzioni
- Dipendono dallo stato emotivo del soggetto
- La presenza di un vetro fra il soggetto e il dispositivo rende inefficace l'acquisizione.

VOLTO – VANTAGGI vs SVANTAGGI

Per quanto riguarda i vantaggi abbiamo:

▪ **Universalità**, tutti hanno un volto, a differenza delle impronte, è molto difficile modificarlo con abrasioni, a meno di chirurgia;

▪ **Efficienza**, le caratteristiche del volto sono limitate, e quindi la dimensione della chiave di ricerca è molto piccola;

▪ **Efficacia**, è molto performante nella verifica (match 1-1), a differenza però del riconoscimento (match 1-N);

▪ **Misurabilità**;

▪ **Accettabilità**;

▪ **Acquisizione**, costo dei dispositivi basso, e permette acquisizioni attive e passive.

Mentre gli svantaggi sono:

▪ **Unicità**, possono esistere dei sosia con le stesse caratteristiche;

▪ **Efficienza**, non performante nel riconoscimento;

▪ **Stabilità**, difficoltà con la variabilità intraclasses;

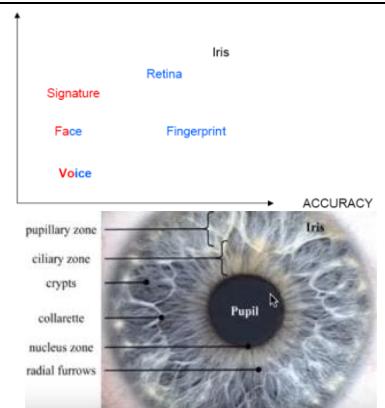
▪ **Insidia**, è abbastanza facile da raggiungere (Physical spoofing), ad esempio con chirurgia o trucco.

8. IRIDE

L'**iride** tra le biometriche è sicuramente quella che garantisce il più basso livello di **False Acceptance Rates (FAR)**, è difficile da replicare ed è abbastanza veloce durante il processo di riconoscimento, infatti per l'iride il problema sostanziale è l'individuazione piuttosto che il riconoscimento. Per quanto riguarda l'iride la permanenza nel tempo è molto robusta in quanto tende a trasformarsi poco (quasi nullo), anche con patologie oculari essi non viene influenzata a differenza della retina. L'iride è una biometria molto costosa sia in termini di costo per le attrezzature sia per l'estrazione delle caratteristiche, ma essa è una delle più accurate, è **contactless** quindi ha bisogno di una detection.

Dal punto di vista anatomico, l'iride è un muscolo avente una funzione cromatica, che agisce sul foro presente al centro dell'occhio, ovvero la pupilla. Quest'ultima è il foro attraverso il quale entra la luce del sole, che va a impattarsi sulla retina. I muscoli che agiscono attraverso movimenti "involontari" hanno il compito di comprimere e far dilatare la pupilla per consentire l'entrata di una certa quantità di luce. Inoltre, dal punto di vista scientifico le **striature** presenti sull'iride (**pattern irregolari**) caratterizzano la cromaticità dell'iride. È possibile individuare tra due consanguinei anche una stessa cromaticità dell'iride, ma ciò che viene diversificato sono appunto tali striature casuali (solchi e creste) che vanno a caratterizzare la parte cromatica.

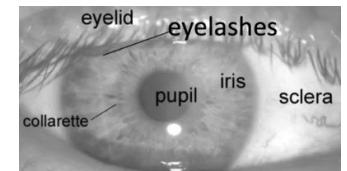
Quindi nell'iride si va a rilevare la **distribuzione** di tali pattern irregolari, piuttosto che la cromaticità del colore, infatti tali pattern rendono unica l'iride dal punto di vista dell'identificazione. Addirittura, l'iride destra è diversa da quella sinistra. L'iride si trova tra la cornea e il cristallino.



- Vantaggi:** l'iride è ben visibile benché protetta dalle palpebre. Non varia nel tempo ed è un tratto biometrico estremamente discriminante rispetto ad altri individui, ma anche all'interno dello stesso individuo. L'immagine può essere acquisita senza avere un contatto diretto (**contactless**). L'acquisizione dell'iride può avvenire in **due modalità** differenti: **infrared** e **nel visibile**.
- Svantaggi:** la superficie sulla quale bisogna poter effettuare l'estrazione delle caratteristiche è molto piccola, nella migliore delle ipotesi si parla di poco più di 3 cm². Per poter acquisire l'iride, si dovrebbe disporre di dispositivi ad elevata risoluzione, che ovviamente dovrebbero adattarsi ad altre situazioni di contorno, ad esempio l'utente non dovrebbe indossare degli occhiali scuri, mentre per altre biometriche una parziale occlusione non ostacola un tentativo di riconoscimento.

L'iride come accennato può essere acquisita sia nel visibile che nell'infrarosso. La differenza sostanziale è legata alla quantità di **melanina**. Ovvero nel caso della **luce visibile**, la melanina assorbe la luce visibile, evidenziando gli strati che compongono l'iride. L'immagine ottenuta con la luce visibile non permette di estrarre tutte le informazioni tridimensionali per una serie di problematiche, ma può essere usata per il **riconoscimento**.

Nel caso della **luce infrarossa**, essa viene utilizzata nelle operazioni di **verifica** (utente altamente collaborativo), in questo caso la melanina riflette la maggior parte della luce infrarossa e assorbe la maggior parte della luce visibile, rendendo limitata la **riflessione**.



La tessitura è più visibile e questo consente un elevato livello di sicurezza soprattutto nel contesto della verifica, appunto infatti si rende più adatta in sistemi biometrici basati sul riconoscimento dell'iride.

Uno dei problemi fondamentali dell'iride è la presenza di una serie di elementi che da un lato possono essere considerati **perturbatori**, e dall'altro elementi **caratteristici** della zona periocularare. In figura sopra si evidenziano tutti gli elementi caratteristici della zona periocularare. Tutti questi elementi possono essere considerati parte di disturbo che possono essere combinate con un'altra serie di tipiche situazioni rumorose per l'acquisizione dell'iride come una scarsa illuminazione, un'immagine con Blurring, un'occlusione o l'angolazione da cui viene effettuato lo scatto può essere sfavorevole per l'acquisizione. Tutte queste caratteristiche tendono a rendere difficile sia il processo di detection e sia il processo di estrazione delle caratteristiche.

Normalmente un algoritmo per estrarre le caratteristiche dell'iride è caratterizzato da una serie di passi:

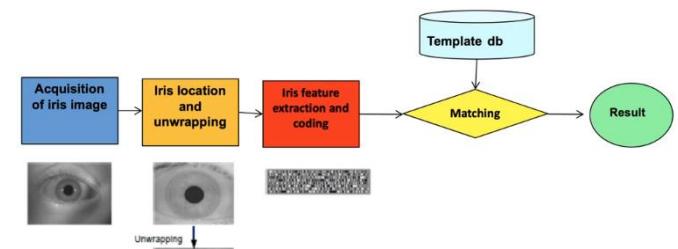
- Segmentazione (fase di detection)**, individuare che all'interno di un'immagine è presente un iride;
- Normalizzazione**, tutti i punti dell'iride vengono trasformati in coordinate polari che permette di avere un vettore a dimensione fissa;
- Coding**, che permette di estrarre le caratteristiche sulla base di un modello diversificato da caso a caso;
- Matching**, dove si confronta il codice prodotto nella fase precedente.

IL PRIMO E PIÙ FAMOSO ALGORITMO – DAUGMAN:

Il primo algoritmo, completamente automatico, utilizzato per l'analisi dell'iride fu prodotto da **Daugman**. Esso propose la prima metodologia automatica basata su un algoritmo piuttosto efficiente di **Iris Location (detection)**. Perché il problema fondamentale nell'iride è legato alla necessità di **individuare** un'iride all'interno di una immagine. Pertanto, la detection dell'iride risulta essere fondamentale. Il processo di estrazione delle feature è banale, una volta individuato l'iride, l'estrazione delle caratteristiche è un problema secondario. Se non si inizia il processo di individuazione o lo si sbaglia, tutto ciò che viene dopo risulta essere fortemente compromesso.

Daugman in un primo momento associò l'individuazione di un'iride all'individuazione di **circonferenze** che fossero in grado di determinare sia il **cerchio della pupilla** sia il **cerchio più grande dell'iride**. Quindi due **cerchi concentrici** tra di loro con lo stesso centro ma raggio variabile. Esso immaginò di avere a disposizione un filtro gaussiano, fondamentale per determinare tali cerchi ed eliminare le piccole variazioni (diminuendo i dettagli), per evitare che queste variazioni presenti all'interno dell'iride possano determinare dei **False Alarms**. Immaginò quindi un **Edge Detector** che producesse delle circonferenze a raggio variabile che andavano ad effettuare una convoluzione con un filtro gaussiano, quindi l'immagine veniva lisciata e prodotta poi una circonferenza (edge detector circolare) che individuasse le **transizioni più significative**. Tale algoritmo si fermava quando si raggiungeva il cerchio a raggio massimo. Partendo dalla zona centrale (x_0, y_0) il primo cerchio che veniva prodotto a raggio massimo era quello che determina la transizione **tra la pupilla e l'iride**, e il secondo cerchio a raggio massimo che veniva prodotto era quello che determina la transizione **tra l'iride e la sclera**. Quindi effettuando la convoluzione tra il filtro gaussiano che effettua uno **Smoothing** dell'immagine ed evita la produzione di false Alarms e la generazione di cerchi a dimensione sempre maggiore, si determina questo operatore integro-differenziale che infine determina i candidati ideali a identificare l'iride. Per quanto riguarda la **formula** utilizzata da Daugman, $G_\sigma(r)$ è la gaussiana che pulisce l'immagine e fa sì che ogni volta che viene generato un nuovo cerchio si effettua la convoluzione con tale gaussiana, eliminando le piccole variazioni di disturbo.

Mentre come elemento di convoluzione si ha un integrale curvilineo, ovvero $\frac{I(x,y)}{2\pi r}$, che fa la somma delle intensità, normalizzata con la dimensione del raggio, e le effettua al variare delle varie circonferenze che trova concentriche.



$$\max_{(r,x_0,y_0)} \left| G_\sigma(r)^* \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

Quindi ogni volta, si calcola la circonferenza, si calcola la somma dei pixel all'interno di questa circonferenza e si calcola il valore ottenuto, alla fine si prende il massimo, ovvero la circonferenza il cui raggio ha determinato la massima variazione, e quest'ultima è candidata la prima volta ad essere la circonferenza della pupilla e la seconda volta la circonferenza dell'iride. Tendenzialmente, si va da una zona scura ad una chiara, il primo passaggio si va da pupilla a iride, mentre il secondo si va dall'iride alla sclera, per questo si cerca il massimo, facendo variare il raggio e fissando il valore (x_0, y_0) . In figura si può osservare come opera l'algoritmo di Daugman. In maniera sequenziale a partire da un punto di un ipotetica iride inizia col generare una serie di cerchi che vanno a determinare le transizioni più elevate. Ogni volta questi cerchi vengono combinati con un algoritmo gaussiano che serve ad evitare variazioni non significative. I due cerchi con raggio massimo vengono poi etichettati per creare la **corona circolare** che determina l'area utile sulla quale andare ad effettuare l'estrazione delle caratteristiche. Ovviamente quest'area utile è solo parziale (in figura vi è un'immagine in infrared).



Riassumendo si tratta di un banale filtro circolare, solo che esso lavora in simultanea con un filtro gaussiano onde evitare falsi allarmi nella zone dell'iride, tentando di arrivare fino alla transizione con la sclera in quanto essa rappresenta la variazione massima di transizione tra un'area e l'altra.

NICE – NOISY IRIS CHALLENGE EVALUATION:

NICE fu una competizione per valutare algoritmi che acquisivano iridi in un contesto non collaborativo (con luce visibile e condizioni di movimento). Furono fatte 2 NICE, la prima valutava algoritmi di **detection** dell'iride, mentre la seconda valutava algoritmi di **estrazioni e riconoscimento** delle caratteristiche dell'iride. L'acquisizione delle iridi veniva fatta che le persone effettuavano un percorso dove venivano posizionate delle telecamere con angolazioni e condizioni di illuminazione differenti. L'unica cosa richiesta ai partecipanti era quella di guardare alcuni simboli del percorso, quando gli utenti vedevano quei simboli, la telecamera effettuava l'acquisizione. La valutazione di questi algoritmi veniva fatta:

1. Sia **Alg** l'algoritmo da valutare, che esegue la segmentazione delle regioni prive di rumore dell'iride.
2. Sia $I=\{I_1, \dots, I_n\}$ il set di dati contenente le immagini dell'iride ravvicinata in ingresso.
3. Siano $O=\{O_1, \dots, O_n\}$ le immagini di uscita corrispondenti agli ingressi sopra descritti, tali che $\text{Alg}(I_i)=O_i$.
4. Sia $C=\{C_1, \dots, C_n\}$ le immagini binarie dell'iride classificate manualmente, fornite dal Comitato Organizzatore di NICE 1. Si deve presumere che ogni C_i contenga la perfetta segmentazione dell'iride e il risultato del rilevamento del rumore per l'immagine in ingresso I_i .
5. Tutte le immagini di I , O e C hanno le stesse dimensioni: c colonne e r righe.

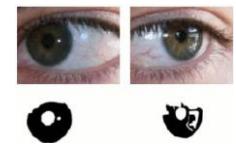
Per il calcolo dell'errore, dovendo confrontare O e C si effettua un OR esclusivo calcolato sul numero di righe e colonne, quindi ogni elemento di O va in XOR ad ogni elemento di C , dato che lo XOR ritorna valore 0 se i due elementi sono uguali, altrimenti 1.

$$E_i = \frac{1}{c \times r} \sum_{c'} \sum_{r'} O(c', r') \otimes C(c', r')$$

L'errore, quindi, viene calcolato come l'errore medio effettuato su ciascuna delle singole immagini, quindi l'algoritmo veniva provato su tutte le immagini producendo il set di immagini O , e ciascun elemento di O veniva confrontato con un elemento di C .

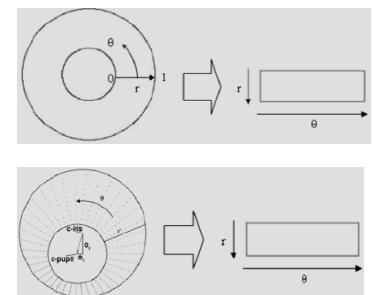
SEGMENTAZIONE GENERALE DELL'IRIDE:

Una volta generato un profilo, si etichettano i **punti neri** come punti appartenenti all'iride e i **punti bianchi** come punti non appartenenti all'iride. Si crea fondamentalmente una maschera che determina quali sono le aree sensibili. All'interno dell'iride ci possono essere dei puntini che vengono esclusi in quanto rappresentano zone di riflesso riducendo l'area dalla quale poter estrarre le caratteristiche. L'output della segmentazione, cioè dell'Iris Detection, è in realtà una **maschera in bianco e nero (immagine binaria)**.



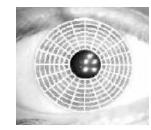
DAUGMAN – IRIS UNWRAPPING [leggere solo]:

Daugman nell'algoritmo aveva previsto una fase di **normalizzazione**, in quanto la dimensione dell'area significativa di un'iride era molto variabile. Essendo molto variabile poteva accadere di dover confrontare due iridi con dimensioni variabili dello spazio utile da cui estrarre le caratteristiche. Poiché l'iride, a differenza dell'impronta digitale, non presenta delle caratteristiche puntuali che vengono estratte ma che viene considerata nella sua intera superficie, vengono considerati tutti i pixel appartenenti alla superficie dell'iride. Si rese necessario che, dato l'intervento di elementi perturbatori, l'iride fosse normalizzato, proprio perché le aree utili erano diverse. Daugman applicò quindi vari algoritmi combinati tra loro che attraverso la combinazione tra coordinate polari e altri elementi di normalizzazione facessero in modo che tutte le superfici delle iridi estratte potessero avere sempre la **stessa dimensione**, creando così un vettore a dimensione fissa indipendentemente dall'area utile dell'iride estratta, proprio per non creare problemi nella fase di matching.



DAUGMAN – ESTRAZIONE DELLE FEATURE [leggere solo]:

Tra gli algoritmi di feature extraction più utilizzati, c'è una variante della trasformata di Fourier basata sui **filtri di Gabor**, si utilizzano dei filtri per catturare alcune informazioni significative dell'iride. Una volta individuata l'area di pertinenza, si tenta di estrarre da essa alcune informazioni aggregate. I filtri di Gabor sono una variante della trasformata di Fourier perché piuttosto che utilizzare la trasformata di Fourier del segnale, utilizzano la trasformata di Fourier combinata con la trasformata di Fourier di una gaussiana.



In realtà il filtro era caratterizzato da una **convoluzione tra la trasformata di Fourier e la trasformata di Fourier di una gaussiana**. Quindi l'immagine viene proiettata nel dominio delle frequenze e non dei pixel. Infine, si avrà un vettore delle caratteristiche che banalmente viene identificato con i coefficienti di Gabor e su di essi viene effettuato un match attraverso la distanza di Hamming.

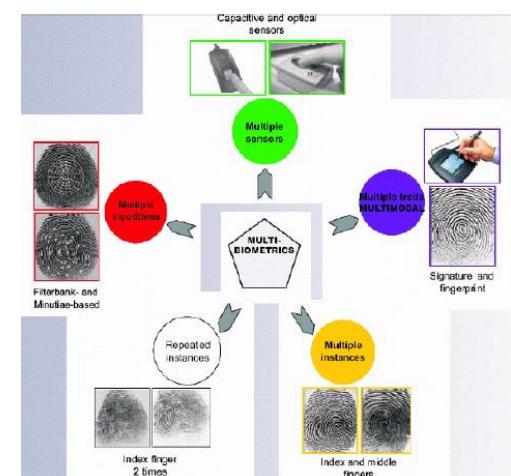
9. SISTEMI MULTIBIOMETRICI

L'idea di sistema **multi-biometrico** nasce dalla necessità di aumentare la **sicurezza** nel contesto di verifica e riconoscimento di una persona. La maggior parte dei sistemi presenti si basa su una **singola biometria**. Ciò li rende vulnerabili a possibili attacchi e scarsamente robusti per una serie di problemi. Un sistema multimodale fornisce una soluzione efficace, poiché gli svantaggi dei singoli sistemi possono essere controbilanciati grazie alla disponibilità di più biometrie. Nello sviluppo di un sistema multimodale bisogna considerare diversi aspetti progettuali:

1. Scelta delle biometrie da utilizzare;
2. Il livello del sistema a cui fare l'integrazione, ovvero **dove** effettuare l'integrazione delle informazioni;
3. La metodologia utilizzata per l'integrazione, ovvero **come** effettuare l'integrazione delle informazioni che verranno poi fuse;
4. Requisiti funzionali, cioè qual è l'hardware che si vorrà utilizzare per l'acquisizione multipla.

Si possono avere vari **tipi di sistemi multi-biometrici**:

1. **Sensori multipli**, ovvero utilizzare una singola biometria ma acquisita con sensori diversi, ad esempio il volto può essere acquisito in modalità termica e poi a luce visibile, perché sensori diversi forniscono caratteristiche diverse;
2. **Tratti multi-biometrici diversi (sistema multimodale)**, sono combinate le caratteristiche di due biometrie diverse, ad esempio impronte digitali e firma;
3. **Istanze multiple**, ovvero la tipologia di biometria è sempre la stessa ma vengono utilizzate più istanze differenti della stessa biometria, cioè si combinano due elementi diversi appartenenti alla stessa classe biometrica (l'impronta di due dita diverse).
4. **Istanza ripetuta**, ovvero l'immagine della stessa biometria viene ripetuta più volte per acquisire un modello più preciso e composito, cioè si combinano più sessioni di acquisizione dello stesso tratto biometrico (più impressioni della stessa impronta).
5. **Algoritmi multipli**, ovvero utilizzare la stessa biometria e lo stesso sensore ma si cambiano gli algoritmi, poiché ogni algoritmo si concentra sull'estrazione e rappresentazione in termini di modelli matematici della biometria si ha una descrizione strutturata differenziata di tale biometria.

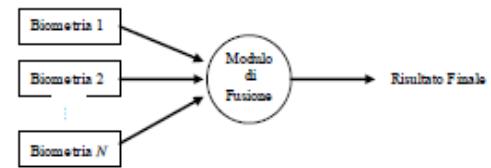


INSIEME DI BIOMETRIE:

La scelta del numero e del tipo di biometria da valutare dipende soprattutto dalla natura del sistema, ovvero dalla **sicurezza** da voler realizzare e dall'**aspetto economico**. Un sistema governativo ha esigenze di sicurezza e di autenticità dell'utente superiori rispetto ad un sistema commerciale, la scelta di una biometria più affidabile quale l'iride o l'impronta digitale è favorita. Un sistema embedded su telefoni o computer palmari privilegia l'uso di biometrie rilevabili con l'hardware in dotazione al dispositivo, quindi voce, volto o firma. Un sistema multimodale può essere progettato secondo tre distinte architetture: in parallelo, in serie e a livello gerarchico (più teorico che pratico).

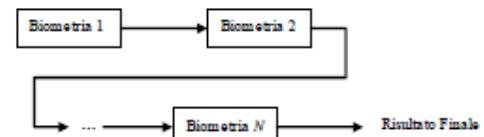
PROGETTAZIONE IN PARALLELO:

Le informazioni estratte dalle molteplici biometrie sono utilizzate simultaneamente per realizzare il riconoscimento. Le acquisizioni sono svolte contemporaneamente e le valutazioni delle singole biometrie sono realizzate indipendentemente e poi combinate mediante opportune tecniche di fusione. Questa tecnica può essere utilizzata per entrambe le modalità di **riconoscimento**, verifica e identificazione. Offre vantaggi maggiori nella prima, in quanto garantisce un accertamento dell'individuo più forte basando la decisione su più parametri di valutazione.



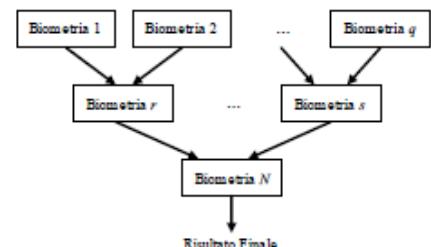
PROGETTAZIONE IN SERIE:

Le valutazioni delle singole biometrie, generalmente due o tre al massimo, sono effettuate in cascata. Questa tecnica può essere utilizzata in entrambe le modalità di riconoscimento. Essa è più vantaggiosa nella modalità di **identificazione**, poiché può essere utilizzata come strumento di filtraggio dei possibili candidati ad ogni processo di confronto. Solitamente si utilizzano diversi tipi di biometrie, una prima di rapido confronto ma di limitata attendibilità per eseguire una prima selezione di una rosa di candidati, sui quali applicare, quindi, una seconda biometria più lenta ma più affidabile per la decisione finale.



PROGETTAZIONE A LIVELLO GERARCHICO:

Si usano metodologie di classificazione individuali per le varie biometrie, quindi i risultati sono memorizzati in una struttura ad albero che facilita la decisione finale. Questa modalità è particolarmente utile nel caso di numerose biometrie da integrare.



LIVELLI E STRATEGIE DI FUZIONE:

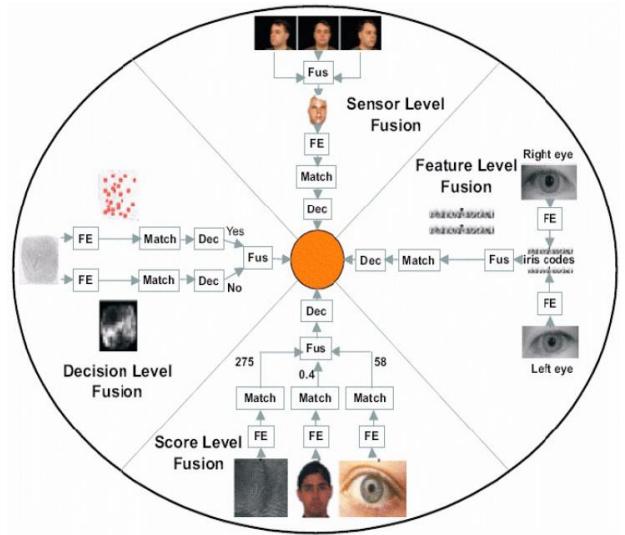
In un sistema biometrico le fasi di Verifica o Identificazioni sono suddivisibili essenzialmente in quattro moduli:

- Modulo di **acquisizione** della specifica biometria;
- Modulo di **estrazione delle caratteristiche** e creazione del template relativo alla biometria rilevata;
- Modulo di **confronto**;
- Modulo di **decisione**.

Una volta decisa quale tipo di soluzione multi-biometrica adottare, resta da stabilire come combinare le informazioni differenziate.

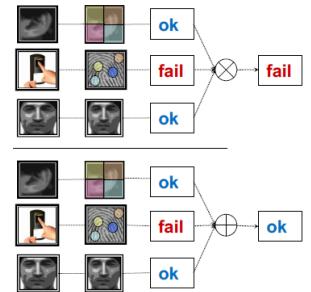
Le tipologie di fusione sono:

1. **Fusione a livello di sensore**, quando viene acquisito il dato, si fondono questi dati creando un unico oggetto "aumentato", ad esempio vengono acquisite più pose di un volto formando un modello tridimensionale. Una volta formato l'oggetto aumentato, si effettua l'estrazione delle caratteristiche, il match ed infine prendere una decisione;
2. **Fusione a livello di caratteristiche**, si acquisisce una biometria (occhio sinistro) e si estraggono le caratteristiche, poi si acquisisce un'altra biometria (occhio destro) e si estraggono le caratteristiche, successivamente si combinano queste caratteristiche (dei due occhi). Quindi avviene una fusione di caratteristiche estratte dalle immagini e poi si procede col match e decisione;
3. **Fusione a livello di score**, le biometrie vengono acquisite separatamente con sensori differenti, successivamente si estraggono le caratteristiche, per ciascuna di queste si prende (sulla base del match) una misura di similarità e queste misure vengono fuse tra loro per poi darlo in input al modulo di decisione;
4. **Fusione a livello decisionale**, vengono estratte le caratteristiche da una o più biometrie, si effettua l'estrazione, il match e la decisione il tutto in modo indipendente, successivamente il risultato della decisione viene dato in input al modulo di fusione.



Sono possibili diverse strategie di combinazione. I più semplici implicano una semplice combinazione logica:

- **Combinazione seriale di AND:**
 - Richiedete l'autenticazione globale.
 - Tutte le decisioni positive.
 - Questo migliora il **FAR**.
- **Combinazione parallela di OR:**
 - L'utente può essere autenticato.
 - Anche per una singola modalità biometrica.
 - Questo migliora **FRR**.
- Un'ulteriore importante regola di fusione a livello decisionale è **Majority Voting**.



POLITICHE DI FUSIONE:

Ritardare la fusione risulta sconveniente, perché:

- Il modulo decisionale riceve solo una minima parte delle informazioni acquisite e conosce solo l'esito del confronto delle singole biometrie;
- Comporta l'implementazione e quindi l'elaborazione di due sottosistemi ognuno dedicato alla specifica biometria.

Conviene fare uso di una strategia di fusione il prima possibile poiché si conserva una ricchezza di informazioni maggiore fornita dai dati acquisiti.

Il template derivante dalla fusione può risultare di dimensioni considerevoli, diventando pesante da gestire dal sistema incidendo così sulle performance. Questa strategia non sempre è attuabile in quanto spesso si lavora con sistemi chiusi nei quali non sono note le relazioni tra i dati acquisiti e i template generati oppure può essere difficile fondere caratteristiche provenienti da biometrie diverse. La scelta più vantaggiosa resta una fusione al modulo di confronto.

STRATEGIE DI FUSIONE:

Al livello decisionale è possibile combinare con una strategia **AND/OR**:

- Richiedere che siano superate tutte le prove biometriche (AND) oppure solo alcune (OR).
- Esempio (Volto-Impronta-Mano): Impronta AND (Volto OR Mano).

Al livello di confronto è possibile utilizzare una strategia di combinazione **pesata**:

- Si associa un peso a ciascuna biometria e quindi anche il risultato dei singoli confronti avranno un peso diverso in fase decisionale
- Altre strategie utilizzano funzioni di combinazione matematiche come la somma, il prodotto, la media, o il massimo o minimo dei risultati dei confronti, queste tecniche necessitano di un ulteriore processo di normalizzazione dei tali risultati