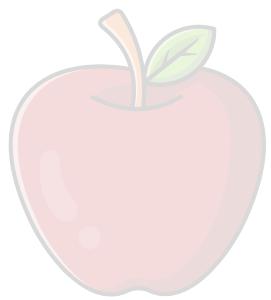


## Biometria

Carmine D'Angelo

A.A. 2022/2023



CoScienze  
Associazione

# Indice

<b>1 Indroduzione</b>	<b>5</b>
1.1 Human Visual System . . . . .	5
1.1.1 Occhio . . . . .	5
1.2 GESTALT Theory . . . . .	5
1.2.1 Principi della psicologia della Gestalt . . . . .	5
1.3 Introduzione alla biometria . . . . .	6
1.3.1 Che cosa è la biometria? . . . . .	6
1.3.2 Caratteristiche biometriche . . . . .	6
1.4 Registramento e verifica di un tizio . . . . .	6
1.4.1 Match esatto e inesatto . . . . .	6
1.5 Architettura di un sistema biometrico . . . . .	7
1.6 Formazione delle biometrie . . . . .	7
1.7 Storia della biometria . . . . .	8
1.8 Le 7 Proprietà di una caratteristica biometrica . . . . .	9
1.9 Tipologia di utente . . . . .	9
1.9.1 7 livelli di autenticazione . . . . .	9
1.10 Misura delle prestazioni . . . . .	9
1.11 Variabilità delle biometrie . . . . .	10
<b>2 Campionamento e Iistogramma</b>	<b>11</b>
2.1 Intensità e luminosità (Brightness) . . . . .	11
2.1.1 Legge di Weber . . . . .	11
2.1.2 Legge di Fechner . . . . .	12
2.1.3 Image Contrast . . . . .	12
2.2 Image Formation . . . . .	12
2.3 Sampling (Campionamento) . . . . .	13
2.4 Quantizzazione . . . . .	13
2.5 Che cosa è un immagine? . . . . .	13
2.6 Risoluzione . . . . .	14
2.6.1 Risoluzione spaziale . . . . .	14
2.6.2 Risoluzione temporale . . . . .	14
2.7 Image Processing . . . . .	15
2.7.1 Image Enhancement (miglioramento dell'immagine): Metodi basati sull'istogramma . . . . .	15
2.7.2 Equalizzazione di un istogramma . . . . .	16
<b>3 Image Filtering</b>	<b>17</b>
3.1 Classificazioni delle trasformazioni . . . . .	17
3.2 Tecniche operanti nel dominio spaziale . . . . .	18
3.3 Tecniche operanti nel dominio spaziale (intorni) . . . . .	18
3.3.1 Filtri spaziali . . . . .	19
3.3.2 Filtro spaziale: La media . . . . .	19
3.3.3 Filtro spaziale: La media adattiva . . . . .	19
3.3.4 Filtro spaziale: La mediana . . . . .	20
3.3.5 Smoothing Gaussiano . . . . .	20
3.3.6 SUSAN: Smoothing Segment Unvalue Assimilating Values . . . . .	20
<b>4 Edge Detection</b>	<b>21</b>
4.1 Edge detection: punti isolati . . . . .	22
4.2 Edge Detection: Linee . . . . .	22
4.3 Edge Detection: Contorni . . . . .	22
4.3.1 Scelta della Soglia (Threshold) . . . . .	23
4.3.2 Perché si usano le derivate? . . . . .	23
4.4 Edge Detection: il gradiente . . . . .	23
4.5 Edge detection: Roberts e Prewitt . . . . .	24
4.6 Sobel Edge detector . . . . .	24
4.7 Edge Detection: Laplaciano . . . . .	24
4.8 Sharpening . . . . .	25
4.9 Canny edge detector . . . . .	25
4.10 Filtri lineari e non lineari . . . . .	25

<b>5 Filtri in frequenza</b>	<b>26</b>
5.1 Definizioni utili . . . . .	26
5.1.1 Sistema lineare . . . . .	26
5.1.2 Linear Shift Invariant System (LSIS) . . . . .	26
5.1.3 Convoluzione nel caso continuo . . . . .	27
5.1.4 Convolution Kernel - Impulse Response (Delta di Dirac) . . . . .	27
5.1.5 Shift-invariant linear System e convoluzione . . . . .	28
5.1.6 Shift-invariant linear system e l'onda sinusoidale . . . . .	28
5.2 La trasformata di Fourier . . . . .	28
5.2.1 Dominio spaziale vs dominio delle frequenze . . . . .	30
5.2.2 Spettro di Fourier . . . . .	30
5.3 Filtraggio con le frequenze . . . . .	31
5.3.1 Tipologie di filtri . . . . .	31
5.3.2 Filtri a passa basso . . . . .	31
5.3.3 Filtri a passa alto (Sharpening) . . . . .	32
5.3.4 Analogie tra dominio spaziale e delle frequenze . . . . .	32
5.3.5 Il teorema della convoluzione . . . . .	32
5.4 Campionamento dei segnali . . . . .	33
5.4.1 Campionamento di un'immagine . . . . .	33
5.4.2 Teorema di Nyquist . . . . .	34
<b>6 Modelli di colore</b>	<b>35</b>
6.1 Colori del CIE . . . . .	35
6.2 Percezione dei colori . . . . .	35
6.3 Modelli di colore . . . . .	35
6.3.1 Conversione tra i modelli di colore . . . . .	36
6.3.2 Relazione tra i modelli di colori . . . . .	36
6.4 Quantizzazione cromatica . . . . .	36
6.5 Palette ottimizzata . . . . .	36
<b>7 Face Detection e Face Recognition</b>	<b>37</b>
7.1 Face Detection . . . . .	37
7.1.1 Struttura di un riconoscitore facciale . . . . .	38
7.1.2 Face localization . . . . .	38
7.1.3 Eye/Mouth localization . . . . .	38
7.1.4 Face contour . . . . .	38
7.2 Face Detection: Viola-Jones . . . . .	39
7.2.1 Integral images . . . . .	40
7.2.2 AdaBoost . . . . .	40
7.3 Face Recognition . . . . .	41
7.3.1 ICA e PCA . . . . .	41
7.3.2 PCA: il metodo . . . . .	42
7.3.3 Eigenfaces - sistemi lineari . . . . .	42
7.3.4 Reti neurali . . . . .	43
7.3.5 Sistemi basati su Grafi . . . . .	43
7.3.6 Termogramma . . . . .	43
7.3.7 Perché utilizzare il volto allora? . . . . .	43
<b>8 Content Based Image Retrieval (CBIR)</b>	<b>44</b>
8.1 Image Features . . . . .	45
8.1.1 Image features: primitive . . . . .	45
8.1.2 Image features: logical . . . . .	46
8.1.3 Image features: abstract . . . . .	47
8.2 Architecture of Visual Information retrieval system . . . . .	47
8.3 Misure per un CBIR . . . . .	47
8.3.1 Recall vs Precision . . . . .	48

<b>9 Noisy Iris Detection and Recognition</b>	<b>49</b>
9.1 Algoritmo di Daugman . . . . .	50
9.1.1 Daugman: iris location . . . . .	50
9.1.2 Daugman : iris unwrapping . . . . .	50
9.1.3 Daugman : feature extraction . . . . .	51
9.2 NICE (Noisy Iris Challenge Evaluation) . . . . .	51
9.2.1 NICE.I . . . . .	51
<b>10 Fingerprint</b>	<b>52</b>
10.1 L'impronta digitale . . . . .	53
10.1.1 Caratteristiche globali di un'impronta digitale . . . . .	53
10.1.2 Caratteristiche locali di un'impronta digitale (minuzie) . . . . .	54
10.1.3 Principali approcci metodologici al confronto di impronte digitali . . . . .	54
10.2 Fingerprint : estrazione dell'impronta . . . . .	54
10.3 Fingerprint : verifica dell'impronta . . . . .	55
10.4 Vantaggi e svantaggi . . . . .	55
10.5 Physical spoofing: Metodo di matsumoto . . . . .	55
<b>11 Multibiometric Systems</b>	<b>56</b>
11.1 Multibiometria . . . . .	56
11.2 Architettura del sistema . . . . .	56
11.3 Livelli e strategie di fusione . . . . .	57
11.3.1 Fusione a livello di sensori . . . . .	57
11.3.2 Fusione a livello di feature . . . . .	57
11.3.3 Fusione a livello di score . . . . .	58
11.3.4 Fusione a livello di decisione . . . . .	58
11.4 Politiche di fusione . . . . .	59
11.5 Strategia di fusione . . . . .	59



**CoScienze**  
Associazione

# Indroduzione

## 1.1 Human Visual System

### 1.1.1 Occhio

L'occhio è una sfera avente un diametro di 20 millimetri e possiede diverse membrane concentriche quali: cornea (trasparente), sclera (opaca), coroide (rete di vasi sanguigni), retina o fovea, coni (circa 6 milioni) che sono sensibili ai colori ed hanno la **Photopic Vision** (capacità dell'occhio di percepire la luce durante il giorno o in presenza di luce), bastoncelli (75/150 milioni) anch'essi sensibili ai colori ed hanno la **Scotopic Vision** (capacità dell'occhio di percepire la luce durante la notte o in presenza di poca luce) e infine abbiamo il cristallino che assorbe l'8% dello spettro visibile (infrarossi e ultravioletti).

## 1.2 GESTALT Theory

La teoria psicologica della Gestalt sostiene che la mente umana organizza le informazioni in modo che le parti si combinino in modi significativi per creare un'esperienza coerente e comprensibile. Secondo la Gestalt, il tutto è maggiore della somma delle sue parti e la mente non elabora le informazioni solo come elementi singoli, ma come un insieme di elementi interconnessi che formano un tutto significativo.

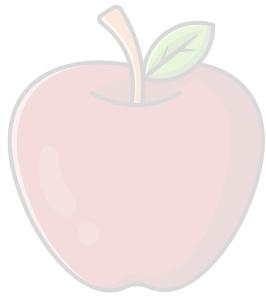


Figura 1.1: Gioca nier automata per la GESTALT

### Sensazione e Percezione

**Sensazione** è il processo attraverso il quale i sensi raccolgono stimoli visivi, uditivi e altri stimoli e li trasmettono al cervello; sono informazioni sensoriali registrate nel cervello ma che non sono state interpretate.

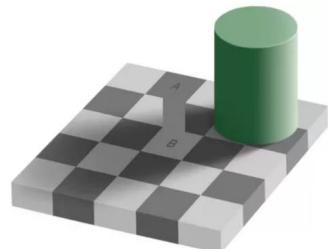
**Percezione** è il processo mediante il quale le informazioni sensoriali sono organizzate e interpretate attivamente dal cervello.

### 1.2.1 Principi della psicologia della Gestalt

- **Figure-Ground:** l'organizzazione dipende da ciò che vediamo come figura (oggetto) e da ciò che percepiamo dal fondo (contesto).
- **(Legge di) prossimità:** gli oggetti si avvicinano nello spazio o nel tempo e sono percepiti come appartenenti ad un solo insieme.
- **(Legge della) somiglianza (similarità):** gli oggetti che hanno caratteristiche simili sono percepiti come unità.
- **(Legge della) continuità:** tendiamo a percepire figure o oggetti come appartenenti ad un insieme se sembrano formare un modello continuo.
- **(Legge della) chiusura:** rileviamo che le figure con delle lacune siano complete.

### Illusioni:

Un'illusione è una distorsione di una percezione sensoriale o cognitiva, causata dal modo in cui il cervello è solito organizzare ed interpretare le informazioni che riceve. Le illusioni possono coinvolgere tutti i sensi, ma quelle ottiche sono le più emblematiche e conosciute, dal momento che la vista spesso prevarica gli altri sensi. Una nota illusione è quella di Adelson, dove due quadrati hanno la stessa tonalità di grigio ma all'occhio umano non sembra così.



## 1.3 Introduzione alla biometria

Il problema che vuole risolvere la biometria è verificare se l'identità dichiarata corrisponde all'identità reale.

**Autenticazione di un soggetto** I processi di autenticazione di un soggetto possono essere molto diversificati. Questo può avvenire attraverso un processo di:

- **Riconoscimento:** una persona riconosce una persona con sistemi e supporti di identificazione come carta d'identità, codice fiscale. Oppure una persona ne riconosce un'altra per conoscenza diretta (ambito familiare);
- **Verifica:** Un sistema informatico verifica una persona utilizzando sistemi e supporti di identificazione come smart card, biometriche, ecc...

Esistono però dei limiti, come il dimenticare un PIN o perdere le chiavi fisiche. I metodi per autenticare (verificare) una persona sono 3:

- **Password** cioè qualcosa che si possiede (smart-card, passaporto, etc...);
- **Conoscenza** cioè qualcosa che conosciamo (password, PIN, etc);
- **Biometria** cioè ciò che siamo. Si possono utilizzare l'iride dell'occhio, le impronte digitali, la forma della mano, etc...

### 1.3.1 Che cosa è la 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.

### 1.3.2 Caratteristiche biometriche

Parlando di caratteristiche biometriche è possibile effettuare una classificazione. Esistono caratteristiche biometriche statiche/-fisiologiche e caratteristiche dinamiche/comportamentali. Le statiche hanno una loro stabilità nel tempo ma soprattutto non dipendono dal comportamento della persona e non possono essere variate sulla base di aspetti emotivi (impronte digitali, mano, volto, occhio, dna). Viceversa, le caratteristiche dinamiche 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 (voce, calligrafia, stile di battitura, andatura).

**Standard X9.84** È uno standard che indica quale tecniche biometriche sono riconosciute. Tra queste ritroviamo riconoscimento delle impronte, dell'iride e della retina, facciale etc...

## 1.4 Registramento e verifica di un tizio

Per la registrazione di un soggetto si cercano di estrarre un insieme di caratteristiche, da queste caratteristiche viene creato un modello (**template**) e viene memorizzato in un database o su un supporto portatile. Può essere effettuato singolarmente o in blocco (**batch enrollment**).

Si parla di verifica quando il confronto è **uno a uno**, conferma l'identità dichiarata da un individuo.

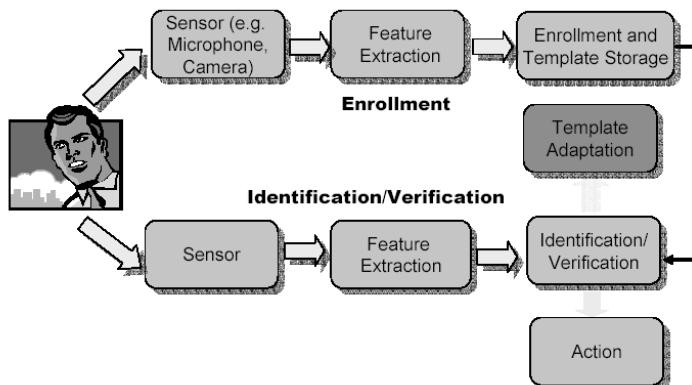
Si parla di riconoscimento quando abbiamo un confronto **uno a molti**, stabilisce l'identità di un soggetto a partire da un insieme di persone registrate.

### 1.4.1 Match esatto e inesatto

- **match esatto** si riferisce a una corrispondenza perfetta tra due campioni biometrici, ad esempio una corrispondenza perfetta tra le impronte digitali o i volti di una persona. In altre parole, tutti i dati biometrici raccolti da due campioni coincidono esattamente, e quindi viene confermata l'identità della persona;
- **match inesatto** si riferisce a una situazione in cui i dati biometrici di due campioni non coincidono esattamente, ma ci sono comunque alcune somiglianze tra i due. Ad esempio, le impronte digitali di una persona possono cambiare leggermente nel corso del tempo, e in questo caso il match potrebbe non essere perfetto ma abbastanza simile per essere considerato una corrispondenza inesatta.

## 1.5 Architettura di un sistema biometrico

### A Generic Biometric System



Un sistema biometrico generalmente è composto da un sensore 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, acquisendo di nuovo i suoi dati e creando un template temporaneo, e verificando che quest'ultimo sia presente nell'archivio per produrre un risultato.

**Registrazione di un soggetto (Enrollment)** : Si associa un insieme di caratteristiche all'identità di un soggetto, raccolgendo 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).

**Verifica e riconoscimento** : Non sempre una stessa biometrica può essere utilizzata sia per la verifica che per il riconoscimento. Dipende dalle condizioni, una delle quali è proprio scegliere se svolgere Verifica o Riconoscimento. Nel caso della 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 (esempio telecamera al gate aeroportuale). In questo caso si acquisisce il dato biometrico e lo si confronta con quelli memorizzati in vari storage. Quindi si stabilisce l'identità di un soggetto a partire da un insieme di persone registrate.

## 1.6 Formazione delle biometrie

Alcune biometriche risultano essere più discriminanti di altre. 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 l'impronta digitale e l'iride. La fase di angiogenesi è il processo fenotipico di sviluppo di nuovi vasi sanguigni a partire da altri già esistenti.

Il **genotipo** si riferisce alla costituzione genetica di un individuo, ovvero all'insieme di tutti i suoi geni ereditati dai suoi genitori.

Il **fenotipo** si riferisce invece alle caratteristiche osservabili di un individuo, come l'altezza, il colore degli occhi, il colore dei capelli, la forma del naso, ecc. Le caratteristiche del fenotipo sono il risultato dell'interazione tra il genotipo e l'ambiente.

In biometria, il genotipo è importante perché influisce sulle caratteristiche del fenotipo che vengono utilizzate per l'identificazione biometrica, come l'impronta digitale, il riconoscimento facciale e la scansione dell'iride

Biometric characteristic	genotypic*	randoypic*	behavioral**
Fingerprint (only minutia)	o	ooo	o
Signature (dynamic)	oo	o	ooo
Facial geometry	ooo	o	o
Iris pattern	o	ooo	o
Retina (Vein structure)	o	ooo	o
Hand geometry	ooo	o	o
Finger geometry	ooo	o	o
Vein structure of the hand	o	ooo	o
Ear form	ooo	o	o
Voice (Tone)	ooo	o	oo
DNA	ooo	o	o
Odor	ooo	o	o
Keyboard Strokes	o	o	ooo

Biometric Trait	Permanence over time
Fingerprint (Minutia)	oooooooooooo
Signature(dynamic)	oooo
Facial structure	oooooo
Iris pattern	oooooooooooo
Retina	oooooooooooo
Hand geometry	oooooooooooo
Finger geometry	oooooooooooo
Vein structure of the back of the hand	oooooooooooo
Ear form	oooooooooooo
Voice (Tone)	ooo
DNA	oooooooooooo
Odor	oooooooooooo?
Keyboard strokes	oooo

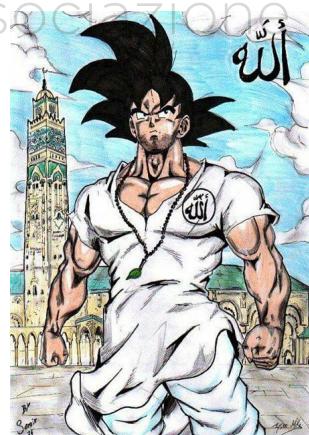
Nella prima tabella in alto (dove più cerchi indicano una corrispondenza forte), vengono analizzate alcune biometriche, e per ognuna di esse vengono individuati tre possibili fattori di caratterizzazione, genotipico, randoypic (o fenotipico) e comportamentale. Concentrandosi sui primi due si può evincere che le componenti genotipiche vengono trasmesse direttamente attraverso il genoma dei genitori. La componente randoypica 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 biometrica, nella seconda tabella. Se una biometrica risulta essere molto variabile, non è pratico utilizzarla per lunghi periodi di tempo. La tabella indica la robustezza delle biometrie, dove il numero di cerchi è relazionato alla robustezza di tale biometrica, ovvero più cerchi più robustezza. Quindi una biometrica se è stata fortemente condizionata da elementi randoypici risulta essere più discriminante.

## 1.7 Storia della biometria

Una prima forma di riconoscimento biometrico possiamo ritrovarla nell'antico Egitto intorno al 2570 AC, durante la costruzione della piramide di Giza. Questa prima forma di biometria venne utilizzata per far sì che gli operai venissero pagati soltanto una volta; come biometria veniva utilizzata l'ossatura umana perché questa a partire dai 20 anni non si modifica più e soprattutto ogni scheletro è diverso (lunghezza avambraccio, lunghezza coscia e gambe, altezza, proporzioni).

Nel tempo l'utilizzo di tecniche biometriche è stato sempre relegato all'ambito forense, fino al 9/11, momento in cui le tecniche biometriche sono iniziate ad essere utilizzate anche per individuare eventuali attentatori, per poi passare negli ultimi anni ad essere utilizzate dalla massa.

L'utilizzo della biometria in un contesto ampio comporta una serie di problemi: utilizzato come strumento di riconoscimento e tracciamento di una persona, può intaccare alcuni aspetti etici e della privacy. Quindi quando si parla di biometria bisogna coniugare più aspetti: Legislazione, applicazioni, implicazioni culturali e sociali e infine tecnologia.



**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:

- **Principi:** L'ossatura umana non si modifica più a partire dai 20 anni, ogni scheletro è diverso;
- **Misure antropometriche:** lunghezza braccio e dita, altezza e larghezza testa e lunghezza piedi;
- Descrizione generale del corpo, caratteristiche mentali e morali, peculiarità della pelle.

Bertillon creò una scheda per classificare i pregiudicati, dove essi venivano classificati sulla base di alcune caratteristiche biometriche descritte poc'anzi. Questa scheda riportava dati che avevano una certa affidabilità, benché nascessero da misurazioni piuttosto variabili dati gli strumenti utilizzati.

Questo sistema 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 nel 1901 e nel 1903 furono registrati due detenuti William West e Will West con misure antropometriche simili.

## 1.8 Le 7 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:** essa deve essere misurabile quantitativamente.
- **Performance (Efficacia/Efficienza):** l'acquisizione, l'estrazione delle caratteristiche e il confronto delle entità non devono essere troppo 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 biometrica);
- **Eludibilità:** la probabilità che il sistema venga raggiirato o indotto in errore deve essere minimizzata.

## 1.9 Tipologia di utente

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).

A seconda della tipologia di utente, cambia la tipologia di progettazione di un sistema biometrico. Un altro aspetto che riguarda l'utente è:

- **Abituato/non abituato:** frequenza con cui gli utenti del sistema si prestano al riconoscimento biometrico (più volte al giorno, settimanalmente, mensilmente, ecc.).
- **Consapevole/inconsapevole:** se l'utente è o no al corrente del processo di riconoscimento in atto.

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

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

### 1.9.1 7 livelli di autenticazione

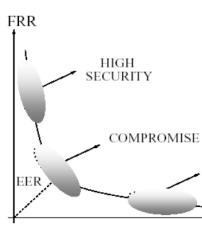
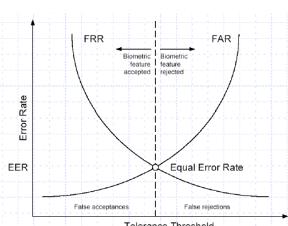
1. Qualcosa che tu conosci;
2. Qualcosa che tu hai;
3. Qualcosa che conosci + qualcosa che tu hai;
4. Qualcosa che tu sei;
5. Qualcosa che tu hai + qualcosa che tu sei;
6. Qualcosa che tu conosci + qualcosa che tu sei;
7. Qualcosa che tu conosci + qualcosa che tu hai + qualcosa che tu sei.

## 1.10 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 di tolleranza è cruciale e dipende fortemente dall'applicazione. 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) – **FAR (False Acceptance Rate)**. FAR = FRR. La scelta più utilizzata è **Equal Error Rate (EER)**.



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 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 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.

## 1.11 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.

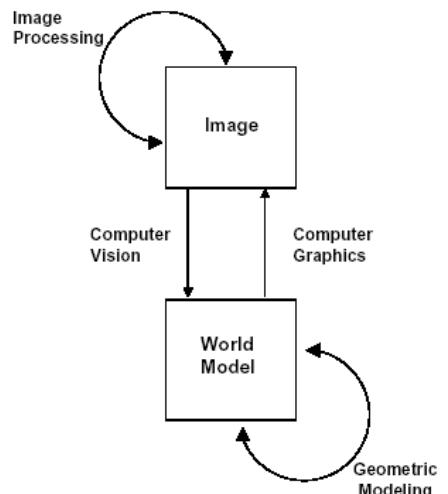
Entrambe le variabilità rappresentano elementi di disturbo per il riconoscimento dell'individuo attraverso tecniche biometriche. La variabilità Intra\_Classe può essere resa variabile: dalla posa, l'espressione, lo scorrere del tempo, l'illuminazione. La variabilità Inter\_Classe, invece, è legata alla presenza di: gemelli, sosia, ecc.



# Campionamento e Istogramma

Alcune definizioni:

- **Image Processing:** è l'applicazione di una qualsiasi tecnica ad un'immagine del mondo reale per ricavarne una nuova che differisce dalla precedente per qualche motivazione (es: radiografia);
- **Computer Vision:** si collega un aspetto di un'immagine del mondo reale a qualcosa di più astratto. È un'evoluzione dell'image processing (trasforma i pixel senza interpretarli), infatti la computer vision assegna ai pixel un significato basato sulle informazioni fornite da un esperto;
- **La Modellazione geometrica:** rappresenta gli oggetti astratti;
- **La Computer Graphics:** prende la rappresentazione astratta di un oggetto e la rende visibile sotto forma di un'immagine.



L'image processing è un'attività di basso livello visto che non va a interpretare ciò che è presente nell'immagine (compressione, rappresentazione e trasmissione). Viceversa la computer vision è considerata di alto livello e sono tutte quelle tecniche che cercano di capire cosa è presente in un'immagine (interpretazione e predizione).

## 2.1 Intensità e luminosità (Brightness)

- **Intensità:** è l'energia luminosa emessa da un singolo punto dell'immagine e dipende dal dispositivo. Ovvero più un dispositivo è in grado in un singolo punto di emettere un'energia luminosa diversa, maggiore è l'intensità prodotta e sarà molto più sofisticato. L'intensità è misurabile.
- **luminosità:** 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 (ed è anche soggettiva), il contesto in cui si trova la rende variabile. La luminosità non è misurabile.

### 2.1.1 Legge di Weber

Weber posizionava gli utenti davanti a uno schermo e tramite un cannone che sparava ad una certa intensità luminosa illuminava lo schermo. Quello che faceva era illuminare con intensità  $L$  lo schermo e con intensità  $L + \delta(L)$  il suo centro. Facendo variare continuamente le due intensità chiedeva all'utente quando riusciva a individuare la variazione d'intensità. Dedusse che quando  $L$  è basso la capacità di distinguere è buona, man mano che  $L$  cresce questa diminuisce.

- Buona distinzione di luminosità:  $\Delta l$  ( $l$  = valore piccolo);
- Cattiva distinzione di luminosità:  $\Delta l$  ( $l$  = valore grande);

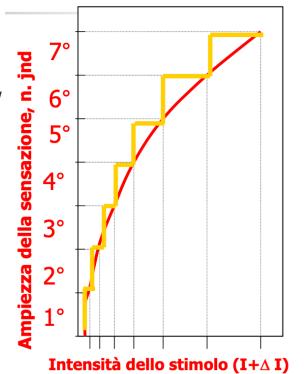
### 2.1.2 Legge di Fechner

La legge di Fechner dice che la percezione umana della sensazione ( $S$ ) aumenta in modo logaritmico all'aumentare dell'intensità ( $I$ ). Questo implica che per raddoppiare la sensazione, l'intensità deve aumentare in modo esponenziale.

Esempio, se l'intensità dello stimolo è aumentata da 10 a 100, la sensazione percepita non sarà aumentata di 10 volte, ma solo di circa 2 volte.

La formula  $S = c \log(I)$  è la formulazione matematica della legge di Fechner. In questa formula,  $S$  rappresenta la sensazione percepita dal soggetto,  $I$  rappresenta l'intensità dello stimolo fisico e  $c$  è una costante di proporzionalità.

- Ascisse: se cresce  $I$ , cresce sempre di più  $I + \Delta I$ ;
- Ordinate: le distanze rimangono uguali, quindi la curva sale meno rapidamente



### 2.1.3 Image Contrast

Il **contrasto** di un'immagine si riferisce alla differenza tra i toni più chiari e i toni più scuri presenti nell'immagine stessa. In altre parole, il contrasto determina quanto le diverse parti dell'immagine si distinguono l'una dall'altra in termini di luminosità e intensità. Il contrasto ( $C$ ) è definito come differenza tra l'intensità in un punto ( $I_p$ ) e l'intensità nel suo intorno ( $I_n$ ) normalizzato con quest'ultimo:



$$C = \left| \frac{I_p - I_n}{I_n} \right| \quad (2.1)$$

- **Alto contrasto:** l'immagine ha una forte differenza tra i toni chiari e i toni scuri, con una distinzione netta tra le diverse parti dell'immagine. Questo significa che i dettagli dell'immagine sono ben definiti e facilmente visibili (non sempre). Le immagini ad alto contrasto spesso hanno colori molto saturi e forti.
- **Basso contrasto:** l'immagine ha una minore differenza tra i toni chiari e i toni scuri, con una distinzione meno netta tra le diverse parti dell'immagine (non sempre). Ciò significa che i dettagli dell'immagine possono apparire sfocati o poco definiti, con una sensazione di morbidezza o di sfumatura. Le immagini a basso contrasto spesso hanno colori meno saturi e più tenui.

## 2.2 Image Formation

Il processo di digitalizzazione di un'immagine risulta essere molto importante in quanto digitalizzando un'immagine o un video, si perde inevitabilmente una parte dell'informazione. Quando si acquisisce un'immagine con una singola sorgente (dispositivo) si passa da una rappresentazione tridimensionale ad una rappresentazione bidimensionale. 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. Quindi possiamo definire la digitalizzazione come una funzione che prende in input le coordinate di un punto:

$$f(x, y) = \text{riflettanza}(x, y) * \text{illuminazione}(x, y) \quad (2.2)$$

La riflettanza è un valore compreso nell'intervallo  $[0, 1]$ , l'illuminazione invece è un valore compreso tra  $[0, \text{inf.}]$ .

La riflettanza misura la capacità di riflettere parte della luce irradiata su una data superficie o materiale.

Quando si vuole acquisire un oggetto tridimensionale questo viene acquisito da un dispositivo il quale all'interno presenta una rete dove in ogni posizione prevede un sensore incaricato di 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** (picture element).

## 2.3 Sampling (Campionamento)

Vogliamo ricostruire l'immagine tramite l'operazione di sampling (campionamento progressivo). Durante il campionamento decidiamo quanti sensori utilizzare per digitalizzare l'immagine, più ne utilizziamo e migliore sarà la qualità dell'immagine acquisita (es: 4x4, 32x32, 128x128). Il campionamento quindi rappresenta quanti pixel avrà l'immagine finale.



Figura 2.1: Lena Söderberg utilizzata per effettuare "studi scientifici" sulle immagini

## 2.4 Quantizzazione

Un'altra fase del processo di digitalizzazione di un'immagine è la fase di quantizzazione. In questa fase andiamo ad indicare ogni sensore quanti colori può produrre:

- 1 bit: immagine bianco o nera
- 8 bit: immagine in scala di grigi (il colore grigio è il miglior colore che un occhio umano può distinguere in tutte le sue sfumature, l'altro colore è il verde).

Più bit si utilizzano più grande sarà l'insieme dei colori e quindi più dettagli saranno visualizzati.

Poiché la fase di digitalizzazione è costituita da campionamento e quantizzazione, se si sbaglia una delle due fasi si compromette il risultato finale, che in situazioni critiche non è accettabile. Quindi quantizzazione e campionamento vanno calibrate in maniera distinta e rigorosa, anche in base al contesto in cui si opera.

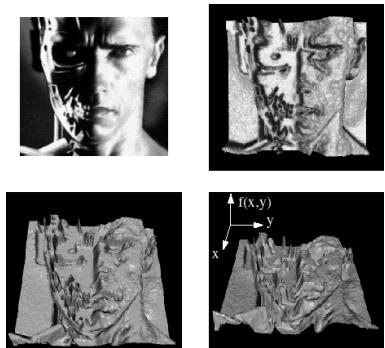
## 2.5 Che cosa è un'immagine?

Un'immagine è un insieme di informazioni luminose che vengono catturate dai sensori. Queste informazioni possono poi essere convertite in informazioni tridimensionali.

Ad ogni posizione  $(x,y)$  dell'immagine corrisponde un valore  $f(x,y)$  che risulta essere il livello di luminosità catturato da un sensore. Inizialmente la rappresentazione tridimensionale di un'immagine può essere rappresentata in termini di una funzione, dove abbiamo i colori più scuri siamo in una zona bassa, man mano che diventano più chiari i colori iniziano ad essere rappresentati come dei rilievi.

Quando abbiamo un'immagine a colori la funzione  $f(x,y)$  è il risultato delle 3 componenti RGB:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix} \quad (2.3)$$



Un'immagine alla fine non è altro che una matrice in cui in ogni cella ritroviamo un valore che corrisponde ad un livello di luminosità variabile a seconda che il sensore assegnato ad un pixel ha catturato un livello di intensità chiaro o scuro.

## 2.6 Risoluzione

La risoluzione si riferisce alla capacità di un sistema di misurare o rappresentare dettagli fini. Nel contesto dell'acquisizione di immagini, la risoluzione dipende sia dal campionamento dell'immagine che dalla sua dimensione fisica.



Figura 2.2: Squall in alta risoluzione

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).

### 2.6.1 Risoluzione spaziale

Quando si cattura un'immagine si potrebbe avere un effetto di blocchetizzazione. Questo può essere dovuto a un basso numero di DPI (dot per inch).

I DPI si riferiscono al numero di pixel presenti in un'area di un pollice quadrato dello schermo, più alto è il numero di DPI, maggiore sarà la risoluzione dell'immagine visualizzata sullo schermo.



Figura 2.3: Squall ti hanno rovinato, addio blocchi

### 2.6.2 Risoluzione temporale

- **FPS** (Frames per Seconds): velocità con cui la GPU completa i frame;
- **Herz (HZ)**: cicli per secondi, associata al display e misura la velocità a cui il display mostra i frame completati;
- **Fluidità**: si riferisce alla capacità di un sistema di acquisizione di immagini (come una videocamera) di catturare dettagli rapidi o in movimento senza perdere la nitidezza o la qualità dell'immagine. Più FPS ci sono più l'immagine risulta fluida;
- **Ghosting** (Latenza del Cambio del Colore dei Pixel): è un effetto visivo che si verifica quando un'immagine in movimento lascia un'impronta residua sullo schermo o sulla superficie di visualizzazione. In altre parole, l'immagine sembra "fantasma" o "trascinare" dietro di sé un'impronta residua.



Figura 2.4: Il più grande esempio di ghosting

- **Tearing**: è un effetto visivo che si verifica quando un'immagine in movimento appare "spezzettata" o "lacerata" durante la riproduzione su uno schermo. Si verifica quando Display e GPU hanno velocità diverse cioè:  $\text{FPS} \neq \text{Hz}$ ;
- **LED RGB**: Si racconta che la presenza di LED RGB aumenti esponenzialmente il numero di FPS.

## 2.7 Image Processing

Quando si effettua un'operazione di Image Processing quello che facciamo è definire 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.

$$g(x, y) = t(f(x, y)) \quad (2.4)$$

oppure applichiamo una trasformazione sulle coordinate dell'immagine (dominio):

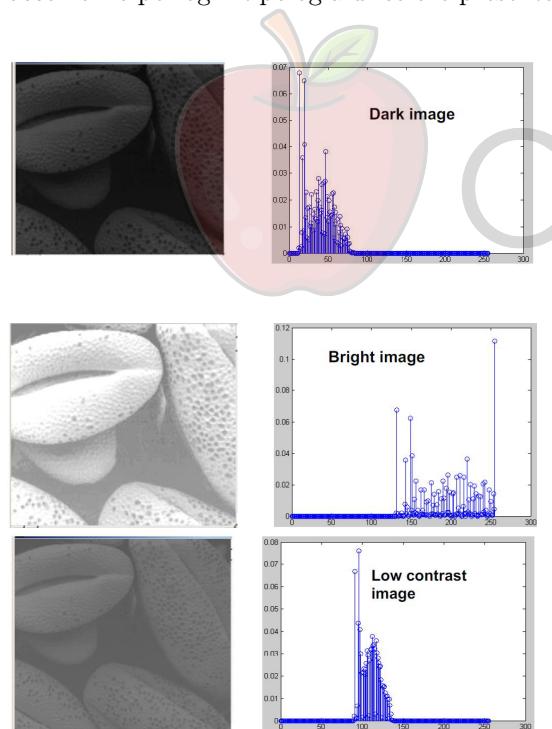
$$g(x, y) = f(t_x(x, y), t_y(x, y)) \quad (2.5)$$

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:** dove oltre al bianco e al 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 complessivamente 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.

### 2.7.1 Image Enhancement (miglioramento dell'immagine): Metodi basati sull'istogramma

Un'immagine può essere anche osservata tramite un'istogramma. Quest'ultimo non è nient'altro che una rappresentazione statistica dell'immagine, attraverso il quale è possibile avere una distribuzione dei suoi livelli di grigio. Si ha il numero di occorrenze per ogni tipologia di colore presente nell'immagine.



Di seguito una immagine con il relativo istogramma ottenuto: 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).

Si ha una prevalenza di colori chiari, molto luminosa, vi sono dei picchi nella parte destra, ovvero nella parte chiara.

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 la seguente immagine (2.5):

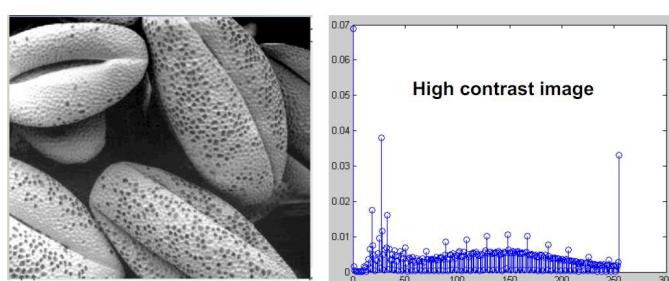


Figura 2.5: Istogramma equalizzato

Quindi un istogramma di un'immagine digitale di  $L$  valori di grigi ( $r$ ) è una funzione discreta definita come il rapporto del numero totale dei pixel con il valore di grigio  $r_k$ :

$$p(r_k) = \frac{n_k}{n} \quad r_0, \dots, r_{L-1} \quad (2.6)$$

dove

- $n_k$ : numero dei pixel con il valore di grigio  $r_k$
- $n$ : numero totali di pixel dell'immagine

### 2.7.2 Equalizzazione di un istogramma

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 218 punti (numero dei campioni). Questo significa che, detto  $H(i)$  l'istogramma (somma dei valori sull'asse delle ordinate), deve valere che:

$$\sum_{i=0}^{255} H(i) = 2^{18} \quad (2.7)$$

Se si vuol rendere l'istogramma piatto significa che si vuole ottenere per ogni livello:

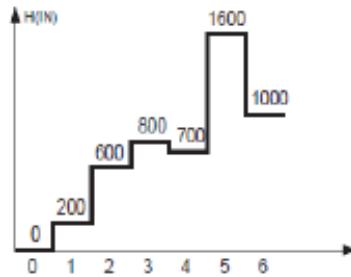
$$H(i) = \frac{2^{18}}{256} = 2^{10} = 1024 \quad \forall i \quad (2.8)$$

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 vogliamo  $1024 + 1024 = 2048$  punti. Poiché nel primo ne sono già stati inseriti 800 si dovrà cercare di approssimare, per il secondo, il valore di 1248 (2048 - 800). 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 ancora una volta vogliamo 1024, 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 3.

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



Quindi con l'equalizzazione di un istogramma quello che vogliamo ottenere è che un istogramma iniziale  $G$  venga trasformato in un altro istogramma detto  $\hat{G}$ . Quindi:

$$\hat{G} = T(G) \quad (2.9)$$

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  $\hat{G}$ . 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".

# Image Filtering

Filtrare un'immagine significa trattenere delle informazioni o eliminarle tramite l'uso di una trasformata. In generale una trasformazione dell'immagine è caratterizzata nel seguente modo:

$$g(x, y) = T[f(x, y)] \quad (3.1)$$

dove:

- $f$  è l'immagine in 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.

## 3.1 Classificazioni delle trasformazioni

Le trasformazioni che possono essere applicate ad un'immagine/video possono essere classificate nel seguente modo:

- **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 agiscono su una trasformazione dei pixel su un concetto diverso. Come si può intuire, tali tecniche implicano un costo computazionale aggiuntivo compensato da una qualità della trasformazione dell'immagine migliore.
- **Tecniche ibride:** ovvero una combinazione di varie tecniche.

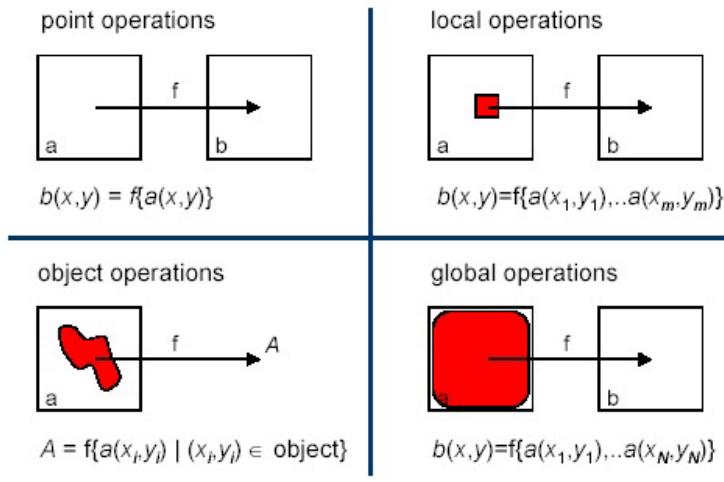
Possiamo suddividere le tecniche delle trasformazioni in ulteriori 4 famiglie:

- **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.  $b(x, y) = f(a(x, y))$
- **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.  $b(x, y) = f(a(x_1, y_1), \dots, a(x_m, y_m))$ . Normalmente utilizzate per "edge detection" o "blur"<sup>1</sup> dell'immagine.
- **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.  $A = f(a(x_i, y_i) | (x_i, y_i) \in \text{object})$
- **Global Operations:** ogni pixel dell'immagine di destinazione, dipende da tutti i pixel dell'immagine di partenza.  $b(x, y) = f\{a(x_1, y_1) \dots a(x_N, y_N)\}$

### Alcune tecniche del point processing

- **Originale:**  $x \rightarrow x$ ;
- **Oscure:**  $x \rightarrow x - 128$ ;
- **Basso contrasto:**  $x \rightarrow x/2$ ;
- **Basso contrasto non lineare:**  $x \rightarrow ((x / 255.0) \wedge 0.33) * 255.0$ ;
- **Inversione:**  $x \rightarrow 255 - x$ ;
- **Lighten:**  $x \rightarrow x + 128$ ;
- **Aumento del contrasto:**  $x \rightarrow x^2$ ;
- **Aumento del contrasto non lineare:**  $x \rightarrow ((x / 255.0) \wedge 2) * 255.0$

<sup>1</sup>sfocatura per eliminare dei dettagli in un'immagine



## 3.2 Tecniche operanti nel dominio spaziale

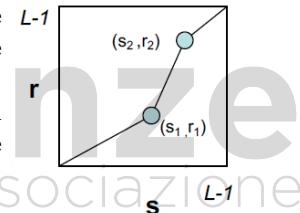
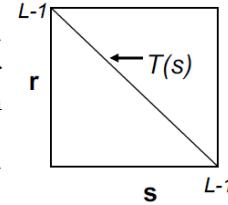
Partendo dal point processing analizzeremo alcune trasformazioni.

Nei grafici di fianco sull'asse delle ascisse abbiamo i livelli di grigio dell'immagine di partenza,  $r$  invece rappresenta un valore di output, mentre  $T$  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  $T(s)$ .

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 (Figura 2) ( $T$  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.



**Binarizzazione** : è un tipo di trasformazione che individua una soglia (es:110), da quel momento tutti i colori da 0 alla soglia diventano di nero, mentre tutti i colori dalla soglia a 255 diventano di colore bianco.

**Negativo** : Produrre il negativo di un'immagine, significa associare ad ogni pixel il livello di grigio complementare.

## 3.3 Tecniche operanti nel dominio spaziale (intorni)

La trasformazione  $T$  utilizzata è definita in un intorno  $(x,y)$ .

Si utilizzano i filtri spaziali (mask processing) per effettuare varie tipologie di operazioni.

Si possono individuare tre categorie di filtri che si andranno ad applicare alle immagini:

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

### 3.3.1 Filtri spaziali

Questi filtri rientrano nella categoria dei **Local Operations** dove ogni pixel di destinazione dipende dal pixel di partenza più i pixel nel suo intorno.

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

$W_1$	$W_2$	$W_3$
$W_4$	$W_5$	$W_6$
$W_7$	$W_8$	$W_9$

$$R = w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + w_7z_7 + w_8z_8 + w_9z_9 \quad (3.2)$$

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

Questa classe di filtri è utilizzata per costruire i cosiddetti **Smoothing Filter**, cioè i filtri utilizzati nelle operazioni di pre-processing. 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 o nella fase di digitalizzazione oppure durante la fase di trasmissione. Questi filtri ricadono nella categoria dei **filtri lowpass** (passo bassa) 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 e non lineari. Il primo filtro che verrà affrontato è il Filtro Media (aritmetica) il quale fa parte dei filtri lineari. I filtri non lineari invece, sono il Filtro Mediana, Filtro Max e Filtro Min.

### 3.3.2 Filtro spaziale: La media

Tale filtro non differisce dal concetto di media aritmetica. Quindi ogni pixel dell'immagine di destinazione viene sostituito con la media dei pixel appartenenti ad un'area ben definita dell'immagine originale. Piuttosto che prendere soltanto il pixel  $(x, y)$  nell'immagine input, viene presa 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. Abbiamo:

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

$$g(x, y) = \frac{1}{M} \sum_{f(n,m) \in S} f(n, m) \quad (3.3)$$

Quello che otteniamo è la seguente matrice:  $\frac{1}{9} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  L'applicazione del filtro media produce un'immagine sfocata, questo

può essere utile nel caso in cui si vuole eliminare del rumore. La dimensione della matrice utilizzata ha sempre un impatto e non è detto che questo sia positivo, infatti eliminare sempre più rumore potrebbe portare a sfocare sempre di più l'immagine. Infatti Rumore-contrasto-dettagli sono legati tra di loro, e possono influenzarsi se lavoriamo nel dominio spaziale.

### 3.3.3 Filtro spaziale: La media adattiva

Si può provare a migliorare il filtro media per ottenere una qualità migliore dell'immagine. Un miglioramento è la media adattiva dove si cerca di risolvere la problematica della confusione dei contorni.

Ogni qual volta essa viene utilizzata effettua un confronto tra il pixel  $(x, y)$  e tutti i pixel che sono nel suo intorno. In presenza di piccole variazioni si applica la media, ma in presenza di grandi variazioni si lasciano invariati i pixel di riferimento. Essa infatti lavora sui bordi cercando di preservarli.

- S: intorno del pixel  $(x,y)$ ;
- M: numero di pixel nell'intorno S;
- T: valore di soglia predefinito.

$$g(x, y) = \frac{1}{M} \sum_{f(n,m) \in S} f(n, m) \quad (3.4)$$

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(n,m) \in S} f(n, m) & \text{if } \frac{1}{M} \sum_{(n,m) \in S} |f(n, m) - f(x, y)| < T \\ 2x - y & \text{otherwise} \end{cases}$$

### 3.3.4 Filtro spaziale: La mediana

Un altro filtro per rimuovere il rumore è quello basato sul concetto di mediana. Si considerano i pixel di una regione, li si ordina in maniera crescente e poi ci si posiziona sul valore centrale. 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.

$$g(x, y) = \text{mediana}\{f(x, y)\}_{f(x, y) \in S} \quad (3.5)$$

La media e la mediana si possono combinare per ottenere in filtro ibrido, dove andiamo ad effettuare la media su  $k$  valori centrali.

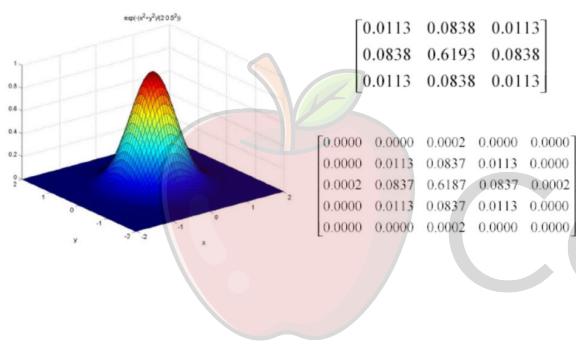
In conclusione possiamo dire che:

- La **media** ha eccessiva velatura(blurring) al crescere della maschera e delle iterazioni con conseguente perdita dei dettagli. preserva con la versione adattiva le forme (scalini);
- La **mediana** riduce il rumore (noise reduction) senza eccessiva perdita dei dettagli (blurring limitato), ma non preserva le forme.

### 3.3.5 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  $N * N$  (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.

Quindi possono essere costruiti nuovi filtri che, fondamentalmente, sono dei filtri media pesati. Mentre prima il peso assegnato dalla media era uguale per tutti quanti, ora si adopera una selezione: più 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. Il filtro gaussiano ha i migliori risultati rispetto agli altri, ma anche questo può essere migliorato.

### Quale filtro scegliere?

Uno dei problemi fondamentali è quello 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.

### 3.3.6 SUSAN: Smoothing Segment Univalue Assimilating Values

È l'evoluzione del filtro gaussiano, il miglioramento sta nel fatto che i pesi non sono stabiliti a priori, ma sono calcolati a **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'area del pixel centrale e si tenderà a dare peso minore ai pixel che si trovano più in lontananza rispetto al centro. Questo calcolo viene effettuato tramite la seguente formula:

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

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 e  $c(r, r_0)$  rappresenta il peso che il pixel in esame ha nel calcolo della media. 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 la grande onerosità in termini computazionali. Il filtro SUSAN preserva gli scalini, risultando essere un filtro ottimale fornendo output migliori.

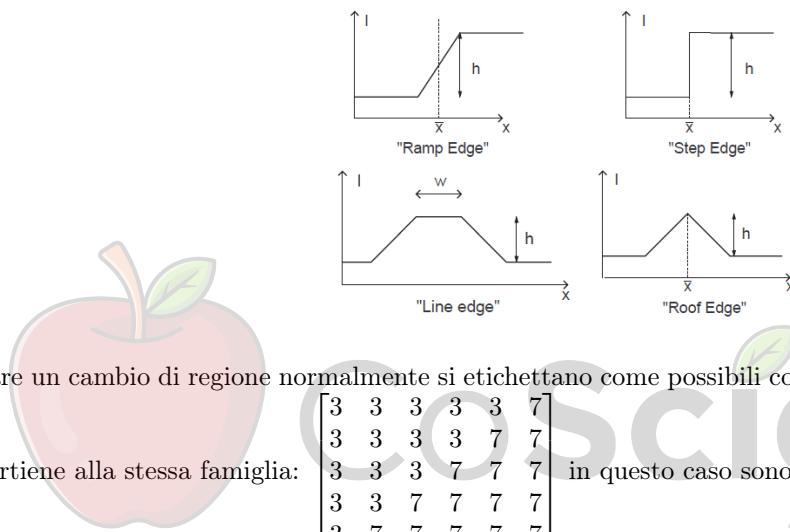
# Edge Detection

Questi tipi di filtri hanno come obiettivo quello di individuare le aree di transizioni tra regioni omogenee. Abbiamo tre tipi di aree di transizione:

- **Punti isolati:** Un singolo punto può rappresentare un contorno, non rappresenta però un'informazione significativa;
- **Linee** (orizzontali, verticali, oblique): insieme di punti con inizio e fine diversa;
- **Contorni:** insieme di punti con inizio e fine coincidenti;

Gli edge 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 allo 0) ad un'area in cui c'è una variazione (o perturbazione). Questa variazione può avere varie forme:

- **Ramp edge:** si passa da una zona uniforme ad un'altra tramite una rampa, avviene in maniera graduale;
- **Step edge:** si passa da una zona uniforme ad un'altra tramite uno scalino;
- **Line edge:** passaggio regione uniforme-rampa-uniforma-rampa discendente- uniforme;
- **Roof edge:** passaggio da una zona uniforme ad un'altra tramite un picco.

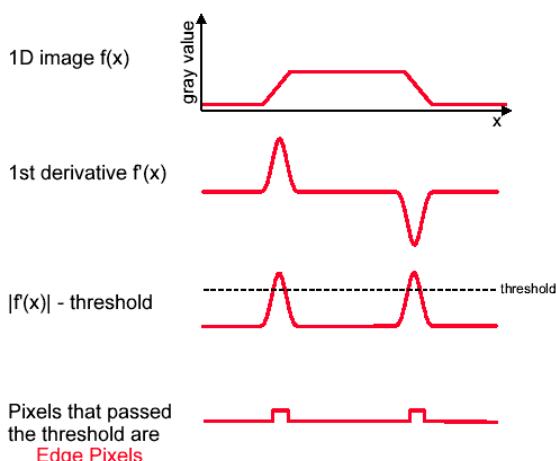


Per individuare un cambio di regione normalmente si etichettano come possibili contorni i punti che hanno almeno un connesso

[	3	3	3	3	3	7
3	3	3	3	7	7	
3	3	3	7	7	7	
3	3	7	7	7	7	
3	7	7	7	7	7	]

che non appartiene alla stessa famiglia: in questo caso sono i 3 e i 7 vicino alla diagonale della matrice.

## Come individuare le variazioni?



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. Viene calcolata la derivata prima per verificare la presenza di variazioni. Nell'immagine la derivata prima risulta essere uguale a zero quando non vi sono valori significativi e risulta avere una variazione laddove esiste un cambio dei livelli. Ritorna ad esser costante in presenza della zona uniforme, per poi trovare un'altra variazione e terminare in modo uniforme. Quindi i punti di massimo e minimo individuati con la derivata prima possono essere significativi, e possono essere utilizzati per capire se c'è stato un passaggio da un'area ad un'altra. Questi pixel presi singolarmente o in modo aggregato sono quelli che possono potenzialmente etichettati come pixel di transizione ovvero **Edge pixels**.

## 4.1 Edge detection: punti isolati

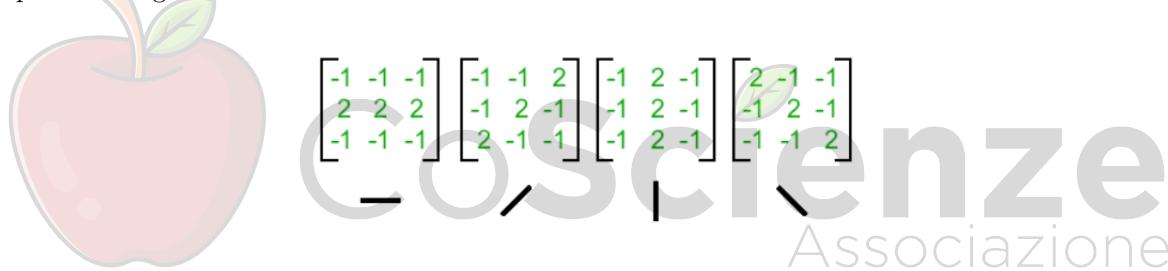
I punti isolati sono le discontinuità più semplici da individuare, in quanto non contribuiscono a definire la morfologia di un oggetto. Il punto corrispondente alla posizione centrale della maschera può essere considerato un punto isolato quando, per esempio con riferimento alla maschera mostrata si 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 una differenza evidente, non si può etichettare questo come punto isolato. Un possibile filtro di edge detector basato sulla derivata prima può sfruttare la maschera  $3 \times 3$ ; quest'ultima presenta valori negativi e un valore positivo centrale, la somma di tutti i 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. Il principio portante di tali maschere, come detto, 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.  $\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

- **Large positive values:** punto chiaro su sfondo nero;
- **Large negative values:** punto scuro su sfondo chiaro.

## 4.2 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.



Anche loro hanno la caratteristica di 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.

- Il primo individua punti di contorno che hanno uno sviluppo orizzontale;
- il secondo invece punti di contorno che hanno uno sviluppo prevalentemente obliquo;
- il terzo punti di contorno che hanno uno sviluppo prevalentemente verticale;
- il quarto individua punti di contorno che hanno uno sviluppo obliquo ma con un senso rispetto al secondo

Si supponga di passare tutte e quattro le maschere su un'immagine, e che in un certo punto si avranno  $R_1, R_2, R_3, R_4$  risposte, diverse, dalle 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.

## 4.3 Edge Detection: Contorni

L'estrazione dei contorni (edge) è molto importante nell'image processing. Il contorno di un oggetto rappresenta infatti 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 di un oggetto. Un edge si presenta in un'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, rispettando poche regole. Infatti, si prende una mascherina di dimensione dispari  $3 \times 3$ ,  $5 \times 5$ , 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.

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).

### 4.3.1 Scelta della Soglia (Threshold)

Uno dei problemi che si presenta è quello di stabilire se è necessario introdurre una soglia. L'introduzione di una soglia è una delle problematiche più importanti nelle fasi di costruzione degli edge detector.

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.

- **Low Threshold:** si individuano come potenziali pixel di contorno anche pixel che hanno variazioni piuttosto piccole. Possono essere introdotti come pixel di contorno anche pixel che non lo sono;
- **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;
- **Hysteresis:** una soglia intermedia dove vengono presi tutti i valori compresi tra una soglia alta e una soglia bassa. Se il pixel individuato è nella soglia alta è di contorno, se è nella zona bassa non lo è, se è in mezzo alle due soglie allora viene etichettato come pixel di contorno solo se collegato con un precedente pixel di contorno. Il calcolo di queste soglie porta ad un aggravio computazionale maggiore.

### 4.3.2 Perché si usano le derivate?

- **Derivata prima:** 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;
- **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.

Mentre la derivata prima indica la presenza di un massimo, la derivata seconda fornisce l'esatta posizione del massimo, nella posizione in cui il rapporto incrementale smette di crescere per iniziare la decrescita.

Quindi il valore della derivata prima può essere utilizzato per determinare la presenza di contorni in un'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.

## 4.4 Edge Detection: il 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 nel seguente modo:

$$\text{Gradiente : } \nabla f(x, y) = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix} \quad (4.1)$$

Del gradiente è possibile calcolare due informazioni: l'entità del gradiente (la grandezza della variazione (magnitude)) e la direzione del gradiente, ovvero in che direzione si muove il contorno:

$$\begin{aligned} \text{Gradient Magnitude : } & \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2} \\ \text{Gradient Direction : } & \tan^{-1}\left(\frac{\delta f}{\delta y} / \frac{\delta f}{\delta x}\right) \end{aligned}$$

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.

## 4.5 Edge detection: Roberts e Prewitt

Sono due edge detector molto sensibili al rumore.

- Roberts Edge Detector Il Roberts Edge Detector utilizza due kernel di convoluzione, o maschere, di dimensioni 2x2, che vengono applicate all'immagine per calcolare la derivata in direzione orizzontale e verticale. I valori dei pixel risultanti vengono poi combinati per calcolare l'intensità del gradiente in ogni punto dell'immagine.

$$F_A = f(x, y) - f(x-1, y-1)$$

$$F_B = f(x-1, y) - f(x, y-1)$$

$$A = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

- Prewit Edge Detector:

Il filtro o operatore di Prewitt è un operatore differenziale che utilizza l'operatore di convoluzione per calcolare le derivate parziali dell'immagine e ottenere il gradiente dell'immagine. È composto da due matrici di convoluzione, una per la derivata parziale rispetto all'asse x e una per la derivata parziale rispetto all'asse y.

$$A = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

## 4.6 Sobel Edge detector

Il filtro o operatore di Sobel è un operatore differenziale che utilizza l'operatore di convoluzione per calcolare le derivate parziali dell'immagine e ottenere il gradiente dell'immagine, tenendo conto della distanza dei pixel dal centro in maniera approssimativa. È composto da due matrici di convoluzione, una per la derivata parziale rispetto all'asse x e una per la derivata parziale rispetto all'asse y.

$$A = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

### Prewit vs Sobel

Anche se il filtro di Prewitt è meno comune rispetto al filtro di Sobel, ha una caratteristica interessante: le due matrici di convoluzione sono simmetriche rispetto all'asse centrale, il che significa che il filtro di Prewitt non introduce alcuna distorsione direzionale nel gradiente dell'immagine, a differenza del filtro di Sobel. Questo può essere un vantaggio in alcune applicazioni specifiche, come nella rilevazione di linee orizzontali e verticali. Tuttavia, il filtro di Sobel è generalmente preferito in molte applicazioni perché è più sensibile ai bordi obliqui e perché fornisce una stima più accurata della direzione del gradiente.

## 4.7 Edge Detection: Laplaciano

Il Laplaciano è una rappresentazione 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 poi i coefficienti che si andranno a riprodurre all'interno della matrice. Questa rappresentazione del Laplaciano è la rappresentazione discreta di una derivata seconda.

È un'approssimazione della derivata seconda orizzontale:

$$\frac{\delta f^2(x, y)}{\delta^2 x} = f'(x+1, y) - f'(x, y) = [f(x+1, y)] - [f(x, y)] - [f(x, y) - f(x-1, y)] = f(x+1, y) - 2f(x, y) + f(x-1, y) \quad (4.2)$$

Si utilizza come maschera quindi:  $(1 \quad -2 \quad 1)$

Mentre se si approssima la derivata in senso verticale si utilizza come maschera:  $\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$

L'operatore Laplaciano è il seguente:  $\nabla = \frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2}$

La maschera utilizzata normalmente è:  $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$  Questo edge detector ha una versione normalizzata, riferito agli 8 vicini, quindi con le differenze delle derivate prime mediate su tre righe e tre colonne, la maschera è:

$$H = \frac{1}{8} \begin{pmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{pmatrix} \quad (4.3)$$

## 4.8 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 un'immagine in bianco e nero ma, con le operazioni di Sharpening, dopo aver estratto i contorni essi vengono sovrapposti all'immagine, per renderli ancor più evidenti. È un filtro highpass.

$$\frac{1}{9} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (4.4)$$

## 4.9 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 (Hysteresis).

La qualità del risultato dell'operatore Canny dipende dall'ampiezza della gaussiana nella prima fase, dimensione del filtro nella prima fase, la scelta delle due soglie ( $T_1, T_2$ ) per l'isteresi nell'ultima fase.

## 4.10 Filtri lineari e non lineari

- **lineari:** Questi filtri modificano il valore di un pixel in modo lineare, cioè in base ad una combinazione lineare dei valori dei pixel circostanti. Esempi di filtri lineari includono i filtri di media, di Sobel, di Laplace;
- **non lineari:** Questi filtri modificano il valore di un pixel in modo non lineare, cioè utilizzando una funzione che non è una combinazione lineare dei valori dei pixel circostanti. Esempi di filtri non lineari includono i filtri di mediana, di massimo e di minimo.

**Filtri lineari** : Il filtraggio lineare di un'immagine si ottiene attraverso un'operazione chiamata convoluzione. La convoluzione è una **neighborhood operation** in cui ogni pixel di output è la somma pesata dei pixel di input vicini.

**Filtri non lineari** : I filtri non lineari non vengono impiegati nell'elaborazione delle immagini con la stessa frequenza dei filtri lineari, perché in molti casi si basano su euristiche strettamente legate alla particolare applicazione. L'idea generale nel filtraggio di immagini non lineari è che invece di utilizzare la maschera spaziale in un processo di convoluzione, la maschera viene utilizzata per ottenere i valori dei pixel vicini e quindi i meccanismi di ordinamento producono il pixel di output.

# Filtrati in frequenza

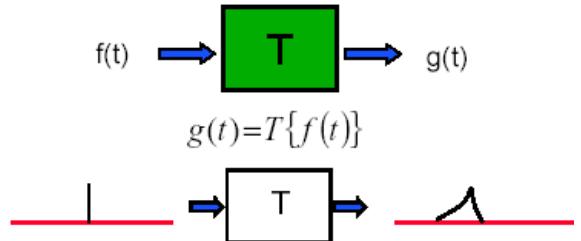
Quando si vogliono applicare dei filtri per rimuovere delle informazioni o per renderle più evidenti, è possibile utilizzare due modalità: operando direttamente sui pixel, oppure trasformare l'immagine in un altro dominio (il vantaggio è la possibilità di ottenere delle informazioni più significative). L'obiettivo è costruire dei filtri che agiscono su altre informazioni correlate con i pixel ma aventi natura diversa. È chiaro che quando si effettua una trasformazione di un'immagine da un dominio all'altro, si deve essere in grado di fare anche l'operazione inversa, riportando l'immagine dal dominio delle frequenze 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 computazionale produce dei risultati qualitativamente migliori.

Si cerca di cambiare dominio perché nel dominio spaziale non riusciamo a ricavare nessuna correlazione fra i pixel vicini (nessun'informazione aggiuntiva), in più i pixel in base al contesto in cui si trovano hanno un significato diverso.

## 5.1 Definizioni utili

### 5.1.1 Sistema lineare

La trasformazione lineare è una trasformazione che agisce su un segnale in ingresso  $f(t)$ , che può essere associato ad un'immagine. Tale segnale entra in una black-box( $T$ , una trasformata nel nostro caso) e produce un output  $g(t)$ .



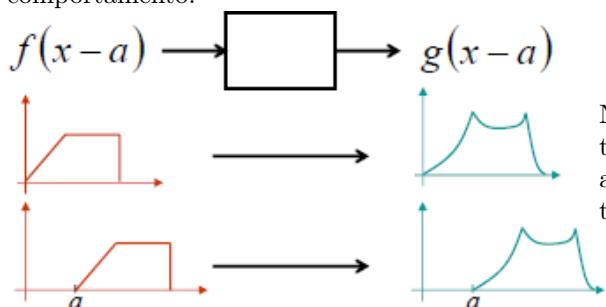
Un sistema per essere lineare deve soddisfare le seguenti due condizioni:

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



### 5.1.2 Linear Shift Invariant System (LSIS)

È una proprietà dei sistemi lineari. Un sistema è shift invariant quando il suo comportamento nel tempo non cambia e l'unica variazione che si ha è la traslazione del risultato. Si può dire che un sistema gode dell'invarianza allo shift se ha il seguente comportamento:



Nell'istante  $t$  il sistema riceve un segnale e riproduce un determinato risultato. Il sistema è **Shift invariant** se traslando  $t$  all'istante  $\alpha$  sull'asse delle ascisse il sistema ricevendo lo stesso segnale, riproduce lo stesso output ma traslato nel tempo.

Di seguito due esempi di sistemi linear shift invariant:

- **Contrast change** della scala di grigi che si estende intorno allo 0:  $T\{f(x)\} = af(x) = g(x)$   
**Shift invariant:**

$$T\{f(x - x_0)\} = af(x - x_0) = g(x - x_0) \quad (5.1)$$

- **Convoluzione:**  $T\{f(x)\} = f(x) * a = g(x)$   
**Shift invariant:**

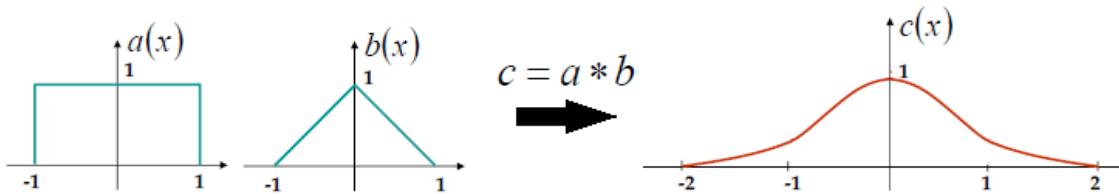
$$T\{f(x - x_0)\} = f(x - x_0) * a = \sum_i f(i - x_0)a(x - i) = \sum_i f(j)a(x - j - x_0) = g(x - x_0) \quad (5.2)$$

### 5.1.3 Convoluzione nel caso continuo

La convoluzione tra due funzioni è un'operazione matematica che combina le due funzioni per produrne una terza. In particolare, la convoluzione si ottiene integrando il prodotto delle due funzioni, spostando una di esse lungo l'asse delle ascisse e calcolando il valore dell'integrale in ogni punto. Matematicamente definiamo quindi la convoluzione di due funzioni  $f$  e  $g$  come segue:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(\alpha)g(x - \alpha)d\alpha \quad (5.3)$$

**NOTA 1:** l'intersezione fra le due funzioni tende ad essere crescente per poi decrescere man mano che l'intersezione termina.



**NOTA 2:** l'operatore matematico della convoluzione è  $*$ .

### 5.1.4 Convolution Kernel - Impulse Response (Delta di Dirac)

L'**impulso unitario** è una funzione matematica che prende anche il nome di **Delta di Dirac**. Permette di rappresentare ogni funzione in relazione al Delta di Dirac.

Il delta di Dirac è definito come segue:

$$\begin{aligned}\delta(x) &= 0 && \text{per } x \neq 0 \\ \delta(x) &= \infty && \text{per } x = 0 \\ \int_{-\infty}^{+\infty} \delta(x)dx &= 1\end{aligned}$$

Le proprietà appena espresse sono per un punto e non per una funzione. Per una funzione  $f(x)$  integrabile abbiamo:

$$\int_{-\infty}^{+\infty} \delta(x)f(x)dx = f(0) \quad (5.4)$$

questa è spesso considerata una definizione alternativa della funzione delta di Dirac. Questa definizione dice che l'integrale di una qualsiasi funzione moltiplicato per una funzione delta di Dirac situata intorno allo zero, ha come valore quello della funzione a zero.

Questo concetto può essere esteso per dare la proprietà Shift invariant ad una funzione  $f(x)$ :

$$\int_{-\infty}^{+\infty} \delta(x - \alpha)f(x)dx = f(\alpha) \quad (5.5)$$

Quindi possiamo anche definire un impulso di segnale come una combinazione di più segnali come segue:

$$\delta[n - k] = \begin{cases} 0 & \text{finché } n \neq k \\ 1 & \text{finché } n = k \end{cases} \quad (5.6)$$

ed ogni segnale può essere rappresentato come una somma lineare di impulsi scalati e shiftati:

$$f[n] = \sum_{j=-\infty}^{+\infty} f[j]\delta[n - j] \quad (5.7)$$

**NOTA:** il delta di dirac utilizzato in una sommatoria ha la funzione di filtro.

### 5.1.5 Shift-invariant linear System e convoluzione

Analizziamo ora cosa è possibile effettuare con i sistemi shift-invariant sfruttando anche la convoluzione.

- $f[n]$ : sequenza in input;
- $g[n]$ : sequenza in output;
- $h[n]$ : risposta del sistema shift-invariant all'impulso unitario:  $h[n] = T\{\delta[n]\}$

Vogliamo mostrare che si può ottenere  $g[n]$ , effettuando una convoluzione fra  $f$  ed  $h$ .

$$\begin{aligned} g[n] &= T\{f[n]\} = T\left\{\sum_{j=-\infty}^{+\infty} f[j]\delta[n-j]\right\} \\ &= \sum_{j=-\infty}^{+\infty} f[j]T\{\delta[n-j]\} \quad (\text{dalla linearità}) \\ &= \sum_{j=-\infty}^{+\infty} f[j]h[n-j] \quad (\text{dalla shift-invariance}) \\ &= f * h \end{aligned}$$

Ottenere  $g$  dalla convoluzione di  $f$  e  $h$  è molto più semplice di effettuare una trasformata. La differenza tra una trasformata e utilizzare il  $\delta$  è che con una trasformata si tende a far sì che funzioni specificamente per la funzione scelta da noi, con il  $\delta$  invece può essere utilizzata per ogni funzione che si vuole adoperare. Per ottenere successivamente l' $h$  piccolo basta effettuare l'anti-trasformata di  $H$  grande e fare poi la convoluzione con  $f$ .

### 5.1.6 Shift-invariant linear system e l'onda sinusoidale

Si vuole costruire una funzione shift-invariant lineare e lo si vuole applicare ad una sinusoida. Questo non modifica la frequenza della funzione ma modifica soltanto **l'ampiezza** (caso che interessa a noi) e la **fase**.

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

**NOTA:** Presa una funzione seno come segue:  $\sin(2\pi\omega x)$ , l'ampiezza è  $\frac{1}{\omega}$ , mentre la frequenza è  $\omega$ .

In sintesi possiamo dire che:

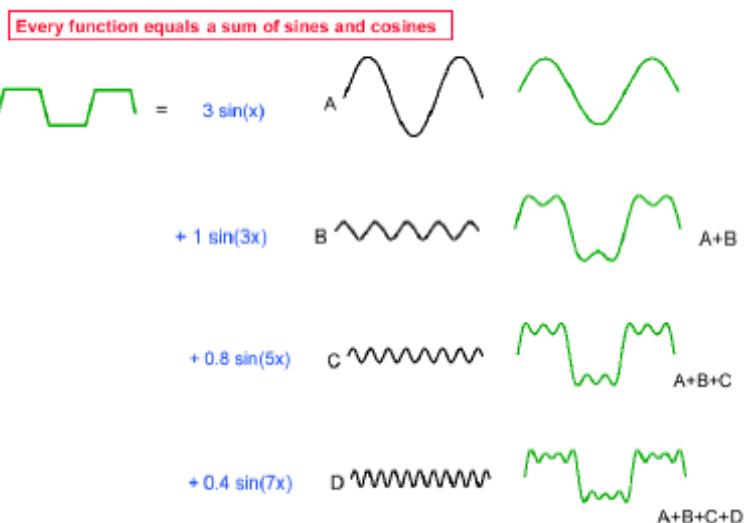
- nel dominio spaziale si fa entrare un impulso e il sistema risponde con una funzione trasformata;
- nel dominio delle frequenze si fa entrare un'onda sinusoidale e il sistema è in grado di modificare la sua ampiezza;
- viene preferita  $H(\omega)$  perché è più semplice da calcolare,  $h(x)$  è più difficile.

## 5.2 La trasformata di Fourier

Fourier affermò che una qualsiasi funzione periodica, può essere scritta sempre sotto-forma di funzione seno e coseno pesate, e con frequenze differenti.

Come si può notare, le varie funzioni sommate tra loro riescono ad approssimare la funzione di partenza. Con infinite frequenze, aumenta maggiormente la frequenza, e ci 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.



La trasformata di Fourier quindi ci dice che è possibile esprimere qualsiasi segnale come somma di sinusoidi shiftate e scalate a frequenze diverse.

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  $\omega$ , dove essa rappresenta le frequenze di  $F(\omega)$  cioè l'ampiezza (ovvero il peso) moltiplicato per  $e^{i2\pi\omega x}$ . Quindi per ogni frequenza  $\omega$ , si assegna un peso stabilito. La trasformata di Fourier permette di prendere un segnale e rappresentarlo in termini di funzioni sinusoidali a frequenza variabile 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 (applicare l'operazione inversa per tornare al dominio originale). Se l'informazione nel dominio spazio-tempo è composta da  $n$  pixel, si avranno  $n$  frequenze nel dominio delle frequenze e viceversa.

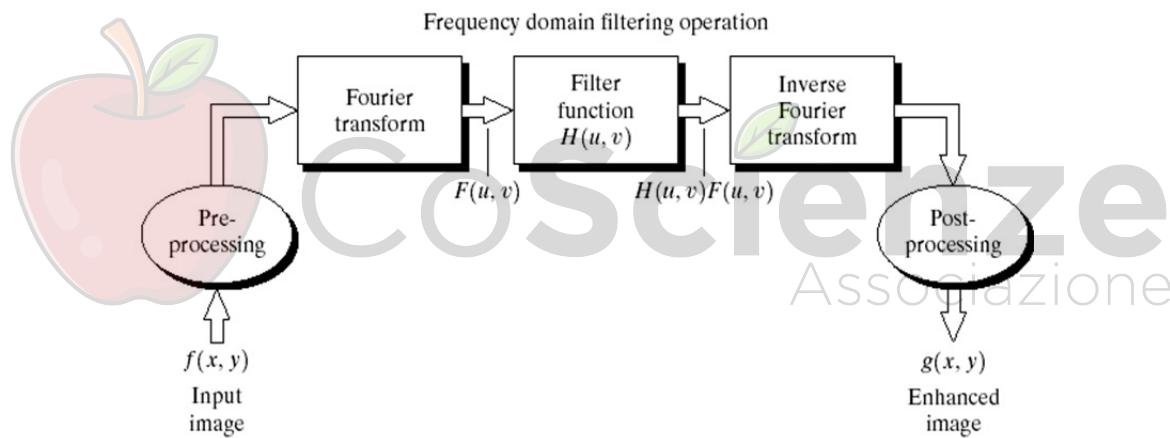
- La trasformata di Fourier trova  $F(\omega)$  dato il segnale  $f(x)$ :

$$f(x) = \int_{-\infty}^{+\infty} F(\omega) e^{i2\pi\omega x} d\omega \quad (5.8)$$

- La trasformata inversa di Fourier presa  $F(\omega)$  ricompone il segnale  $f(x)$ :

$$F(\omega) = \int_x f(x) e^{-i2\pi\omega x} dx \quad (5.9)$$

### Procedimento dell'image filtering nel dominio delle frequenze



Si parte da un'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. 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. Dopo fatto ciò si applica la trasformata inversa per ottenere  $g(x, y)$ . Quindi invece che applicare la trasformazione ad  $f$ , si applica la trasformazione ad una rappresentazione nel dominio delle frequenze di  $f$ .

### Trasformata di Fourier 2D

La Trasformata di Fourier 2D (DFT 2D) è una tecnica matematica utilizzata per analizzare le proprietà delle frequenze di un'immagine. In pratica, la DFT 2D consente di decomporre un'immagine in una serie di componenti sinusoidali, o onde, di differenti frequenze.

- La trasformata di Fourier:

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{i2\pi(ux+vy)} dx dy \quad (5.10)$$

- La trasformata inversa di Fourier:

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{i2\pi(ux+vy)} du dv \quad (5.11)$$

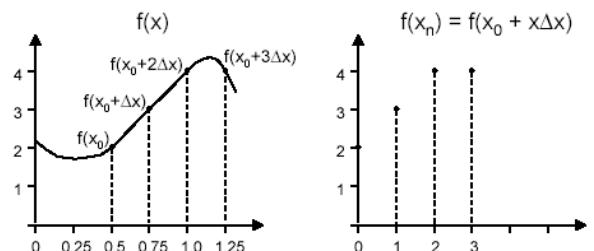
$$f(x) = \sum_{\omega} F(\omega) e^{i2\pi\omega x}$$

or

$$f(x) = \int_{\omega} F(\omega) e^{i2\pi\omega x} d\omega$$

## Trasformata di Fourier discreta

La trasformata di Fourier discreta (DFT) è una tecnica matematica utilizzata per analizzare le proprietà delle frequenze di un segnale discreto. In poche parole, la DFT consente di analizzare la decomposizione di un segnale in una serie di componenti sinusoidali, o onde, di differenti frequenze.



- La trasformata di Fourier discreta è definita nel seguente modo:

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

- La trasformata discreta inversa di Fourier:

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

## Trasformata di Fourier discreta 2D

- La trasformata di Fourier discreta 2D è definita nel seguente modo:

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

con  $u = 0, 1, 2, \dots, N-1$     $v = 0, 1, 2, \dots, M-1$

- La trasformata discreta inversa di Fourier 2D:

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

con  $x = 0, 1, 2, \dots, N-1$     $y = 0, 1, 2, \dots, M-1$

### 5.2.1 Dominio spaziale vs dominio delle frequenze

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 un'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.

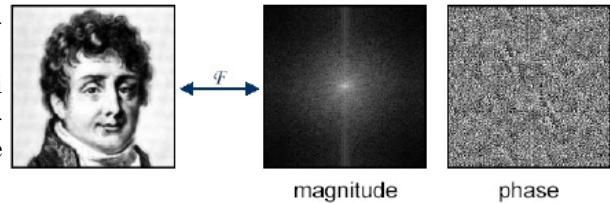
La trasformata di Fourier, applicata ad un'immagine  $N \times N$ , computazionalmente viene a costare  $O(N^2)$ , quindi un aumento di complessità rispetto alla linearità che si verificava nelle trasformate lineari. La **Fast Fourier Transform** costa  $O(n \log n)$ .

### 5.2.2 Spettro di Fourier

Lo spettro di Fourier è una rappresentazione grafica delle componenti di frequenza di un segnale. Consente di analizzare la presenza e l'ampiezza delle diverse componenti di frequenza in un segnale e di visualizzarle graficamente. Lo spettro di Fourier può essere rappresentato in due dimensioni, dove l'asse delle x rappresenta la frequenza e l'asse delle y rappresenta l'ampiezza (può essere rappresentato anche in 3 dimensioni, in questo caso l'asse x = frequenza, y = magnitudine, z = fase).

La Trasformata di Fourier restituisce due componenti, una componente di **magnitudine** e una di **fase**. La magnitudine rappresenta l'ampiezza delle componenti di frequenza, mentre la fase rappresenta lo sfasamento delle componenti di frequenza rispetto ad una fase di riferimento.

La magnitudine è spesso rappresentata graficamente tramite uno spettro di ampiezza, mentre la fase è rappresentata tramite uno spettro di fase. Lo spettro di ampiezza mostra la distribuzione delle ampiezze delle componenti di frequenza nel segnale analizzato, mentre lo spettro di fase mostra la distribuzione degli sfasamenti delle componenti di frequenza.



### 5.3 Filtraggio con le frequenze

Quando vogliamo effettuare una modifica ad un'immagine tramite il dominio delle frequenze quello a cui dobbiamo fare caso sono le basse, medie, alte e altissime frequenze:

- **basse/medie:** utilizzate per la compressione delle immagini o eliminare il rumore;
- **alte:** è dove sono contenuti i dettagli dell'immagine;
- **altissime:** è dove è presente il rumore.

Nell'immagine di fianco vi è la rappresentazione della trasformata di Fourier, dove viene rappresentato lo spettro di Fourier. Nelle ampiezze dello **spettro di Fourier** si può notare 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 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 tale aree vanno eliminate. 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. 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, quindi 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.

Di seguito un altro esempio di taglio delle frequenze in cui si eliminano dei dettagli:

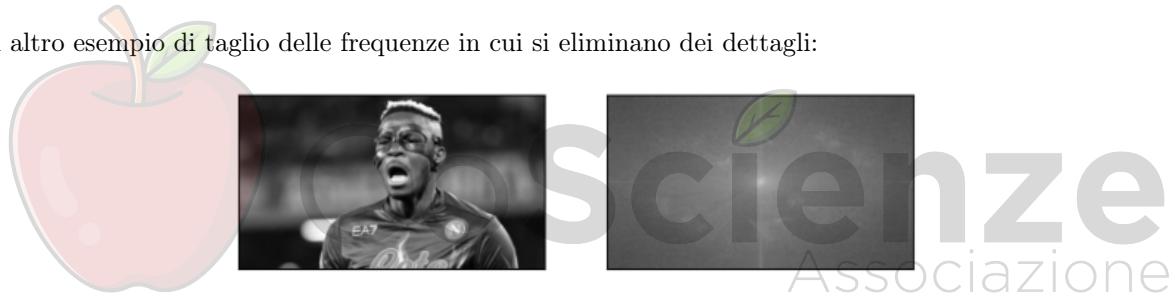


Figura 5.1: Osimhen



#### 5.3.1 Tipologie di filtri

#### 5.3.2 Filtri a passa basso

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 le frequenze che si vogliono tagliare quindi una moltiplicazione per zero. Il filtro diventa quindi un moltiplicatore, attuato sulle frequenze:

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) > D_0 \end{cases} \quad (5.12)$$

$$D(u, v) = \sqrt{u^2 + v^2}$$

$D_0$  = frequenza di taglio, quella rispetto alla quale si stabilisce una soglia.

### 5.3.3 Filtri a passa alto (Sharpening)

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

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{se } D(u, v) > D_0 \end{cases} \quad (5.13)$$

$$D(u,v) = \sqrt{u^2 + v^2}$$

$D_0$  = frequenza di taglio, quella rispetto alla quale si stabilisce una soglia.

### Filtro passa banda

Prende una parte delle basse/medie frequenze e una parte delle alte frequenze:

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 - \frac{w}{2} \\ 1 & \text{se } D_0 - \frac{w}{2} \leq D(u, v) \leq D_0 + \frac{w}{2} \\ 0 & \text{se } D(u, v) > D_0 + \frac{w}{2} \end{cases} \quad (5.14)$$

$$D(u,v) = \sqrt{u^2 + v^2}$$

$D_0$  = frequenza di taglio, quella rispetto alla quale si stabilisce una soglia.

w = larghezza di banda.

**Frequency bands** È possibile calcolare la percentuale di energia che si conserva dell'immagine se si va a costruire un filtro progressivo. Più ci si allontana dalle basse frequenze, minore è il contributo informativo che quelle frequenze danno all'occhio umano. Di norma se si calcola quanta percentuale di energia si conserva, si può notare che si perde un 10% che non influisce sulla qualità dell'immagine.

### 5.3.4 Analogie tra dominio spaziale e delle frequenze

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

Il segnale in input 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 (filtro costruito 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.

### 5.3.5 Il teorema della convoluzione

Il teorema afferma che la convoluzione di due funzioni è uguale al prodotto delle trasformate di Fourier delle due funzioni. Quindi prese due funzioni  $f(x)$  e  $g(x)$ , calcoliamo la convoluzione

$$h(x) = f(x) * g(x) \quad (5.15)$$

allora la trasformata di Fourier di  $h(x)$  è data dal prodotto delle trasformate di Fourier di  $f(x)$  e  $g(x)$ :

$$F[h(x)] = F[f(x)] \cdot F[g(x)] \quad (5.16)$$

Il teorema della convoluzione è anche strettamente legato alla proprietà di commutatività della convoluzione e alla proprietà di commutatività della moltiplicazione nel dominio della trasformata di Fourier. Quindi se invertiamo le operazioni nei due domini anche le operazioni vengono invertite.

Dominio spaziale:

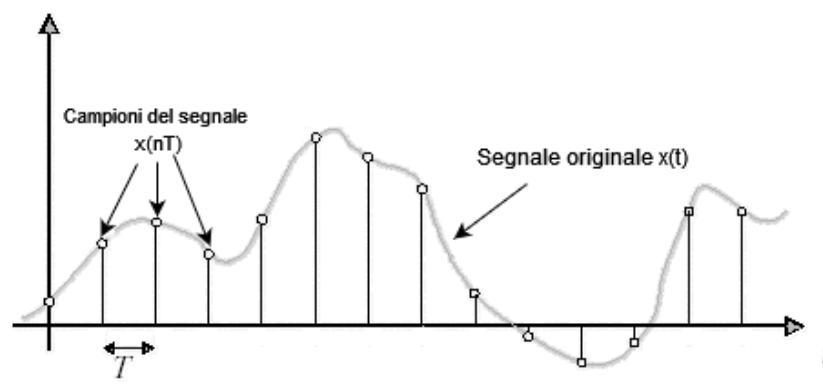
$$h(x) = f(x) \cdot g(x) \quad (5.17)$$

Dominio delle frequenze:

$$F[h(x)] = F[f(x)] * F[g(x)] \quad (5.18)$$

## 5.4 Campionamento dei segnali

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 condizionato dal passo di campionamento. In genere più è ampio il passo di campionamento minore è la qualità del segnale discreto.



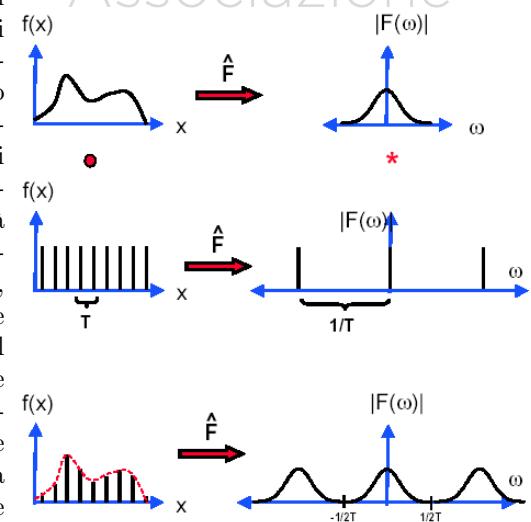
- $T$  è detto **periodo (o passo) di campionamento**;
- $f_e = 1/T$  è detta **frequenza di campionamento**.

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 differente 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 (dominio delle frequenze).

### 5.4.1 Campionamento di un'immagine

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 al passo  $t$  si ottiene nel dominio delle frequenze una serie di impulsi al 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.

Se vogliamo effettuare un campionamento 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 sovrapporsi. Quando le armoniche



sono molte sovrapposte, una parte delle informazioni vanno tra di 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 vadano ad intersecarsi ma che siano contigue. Per stabilire il giusto passo di campionamento, bisogna stabilire un passo di campionamento:  $\frac{1}{T} > 2\omega_{max}$

### 5.4.2 Teorema di Nyquist

Il teorema afferma che per acquisire un segnale analogico in modo da poterlo rappresentare in forma digitale senza perdere informazioni, la frequenza di campionamento deve essere almeno il doppio della frequenza massima presente nel segnale.

Se la frequenza massima di  $f(x)$  è  $\omega_{max}$ , la frequenza di campionamento deve essere maggiore di  $2\omega_{max}$  per ricostruire completamente  $f(x)$  dai suoi campioni. Se la frequenza di campionamento è minore di  $2\omega_{max}$  le repliche sovrapposte producono **aliasing**.

Dalle proprietà della trasformata di Fourier è immediato verificare che il segnale campionato ha come trasformata di Fourier la ripetizione periodica della trasformata  $X(f)$  e del segnale continuo  $x(t)$ , con periodo pari alla frequenza di campionamento  $f_c = \frac{1}{T}$ , moltiplicata per  $f_0 = \frac{1}{T}$ :

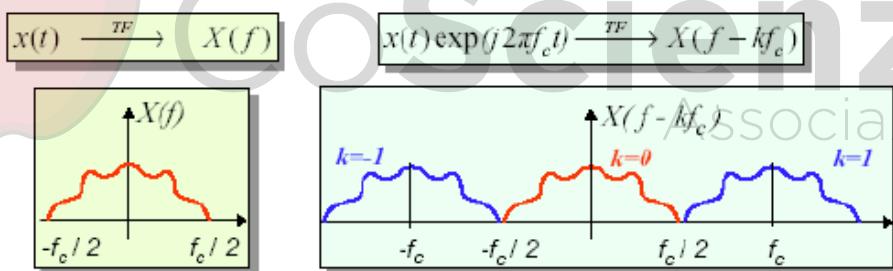
$$X_c(f) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} X(f - \frac{k}{T}) \quad (5.19)$$

Se le infinite repliche traslate in frequenza di  $X(f)$  non si sovrappongono è facile estrarre mediante filtraggio  $X(f)$  da  $X_c(f)$ , cioè riottenere  $x(t)$  da  $x_c(t)$ . Ovviamente ciò richiede una conoscenza a priori della banda B occupata da  $x(t)$ . Tale valore è disponibile, o facilmente misurabile, nella grande maggioranza dei casi di interesse pratico.

#### Teorema del campionamento

Se è noto a priori che il segnale tempo continuo  $x(t)$  non contiene frequenze maggiori di  $\frac{f_c}{2}$  e inferiori a  $-\frac{f_c}{2}$ , esiste un legame univoco tra il segnale continuo nel tempo e i suoi campioni  $x(nT)$ .

Se un segnale  $x(t)$  è campionato con frequenza di campionamento  $f_c$  almeno doppia della massima frequenza contenuta è perfettamente ricostruibile (le repliche in frequenza sono disgiunte). Altrimenti le repliche sono sovrapposte e vi sono frequenze alle quali non è possibile distinguere tra repliche diverse.



# Modelli di colore

L'occhio umano non vede tutti i colori ma possiede uno spettro visibile. Per vedere quali colori l'occhio umano può 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 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. Un oggetto verde 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, ma riporta una sensazione risultante dalla combinazione di tutte le lunghezze d'onda visibili. È inoltre impossibile comunicare la sensazione corrispondente ad un certo stimolo. Quello che possiamo comunicare è che due stimoli diversi producono la stessa sensazione.

## 6.1 Colori del CIE

La CIE si è occupata nel 1931 di classificare i colori, come combinazione di altri chiamati colori primari:

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

Si classificò anche un quarto colore Y (luminanza dello stimolo), ed è la combinazione dei 3 colori primari moltiplicati per alcune soglie:

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

**NOTA:** alcuni colori non possono essere rappresentati come somma dei tre colori primari.

## 6.2 Percezione dei colori

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 e producono una visione **scotopica** (capacità di percepire la luce a bassa intensità) basata solo sulle differenze di luminosità.

## 6.3 Modelli di colore

Abbiamo due classi di colore:

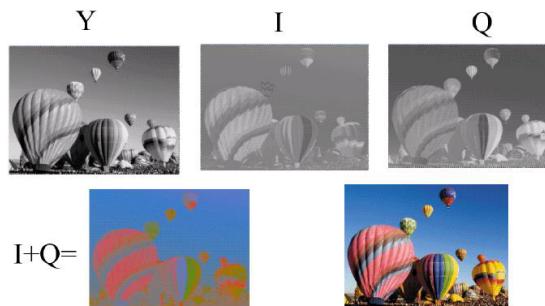
1. **True colors** (colori veri) è ottenuta mediante composizione (sottrattiva o additiva di tre componenti –HSB, RGB, CYM, YIQ–). Ogni componente è quantizzata con un numero definito di bit;
  - **RGB**: è utilizzato per realizzare dispositivi di proiezione quali monitori, TV e nell'elaborazione di immagini. Viene utilizzato anche per immagini satellitari. È un modello **additivo**, si addiziona luce (RGB) al nero;
  - **CYM**: utilizzato per realizzare dispositivi di stampa. È un modello **sottrattivo**, si sottrae luce (CYM) al bianco;
  - **HSV**: è utilizzato nell'Elaborazione di Immagini. È una combinazione di Hue (tonalità), Saturazione (Saturation) e Luminosità (Brightness);
  - **YIQ**: è utilizzato nelle trasmissioni TV e nell'elaborazione di immagini, è anche molto utile per la compressione dei dati. Sfrutta la maggiore sensibilità dell'occhio umano alla luminanza (immagini a livelli di grigio).
2. **Pseudo-colors** (colori falsi) è ottenuta assegnando ad ogni intervallo di colori veri un colore medio.

**Relazione dei colori CMY:** magenta = R + B; ciano = G + B; giallo = R + G. Abbiamo poi 3 possibili combinazioni per il bianco: bianco = blu (primario) + giallo (secondario); bianco = verde (primario) + magenta (secondario); bianco = rosso (primario) + ciano (secondario).

## Relazione dei colori YIQ:

- Y = Luminanza (proporzionale alla quantità di luce percepita dall'occhio);
- I = Inphase;
- Q = Quadrature ( $I + Q =$  informazioni sul colore).

L'occhio umano è più sensibili alla variazione di luminosità che alle variazioni di tono e saturazione.



### 6.3.1 Conversione tra i modelli di colore

$$\begin{bmatrix} C \\ Y \\ M \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

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

### 6.3.2 Relazione tra i 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).

## 6.4 Quantizzazione cromatica

Normalmente le immagini sono quantizzate come segue:

- Immagini Binarie (1 bpp);
- Immagini in scala di grigio (8 bpp, 16 bpp);
- Immagini a colori (16 bpp, 24 bpp, 32 bpp, 48 bpp, 64 bpp);
- Immagini a pseudo colori o falsi colori (8 bpp).

Come la risoluzione spaziale anche la risoluzione cromatica è un compromesso costo prestazioni. Non sempre è necessario avere a disposizione tutti i colori delle immagini true-color (24 bit per pixel, 8 per ogni canale), specie in applicazioni tipiche della grafica. È quindi possibile risparmiare spazio di memorizzazione e ridurre il carico computazionale, riducendo il numero di colori. Per ridurre il numero di livelli di colore:

- Si usano meno bit per banda (es. 5-5-5 o 5-6-5 nel caso dell'hicolor a 15 e 16 bit rispettivamente);
- Si usa una look-up table (mappa di colori): si sceglie un numero finito di colori (es. 256) memorizzati su una tabella ed il valore del pixel è un puntatore a tale tabella che contiene terne RGB.

## 6.5 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.

# Face Detection e Face Recognition

## 7.1 Face Detection

Il volto è considerabile una biometria al limite, in quanto essa è soggetta a cambiamenti temporali, ma è anche la biometrica con la quale si riconoscono le persone. Nel riconoscere un volto esistono due fenomeni la **pareidolia** cioè l'illusione subcosciente che tende a ricondurre a oggetti o profili (naturali o artificiali) un volto; e la **prosopagnosia**, un deficit percettivo acquisito o congenito del sistema nervoso centrale che impedisce a chi ne è colpito di riconoscere correttamente i volti.

Il volto è molto utilizzato per via della sua:

- **affidabilità:** presenta molte caratteristiche uniche e stabili;
- **accettabilità:** non richiede l'utilizzo di strumenti invasivi o fastidiosi, come ad esempio le impronte digitali o la scansione dell'iride.

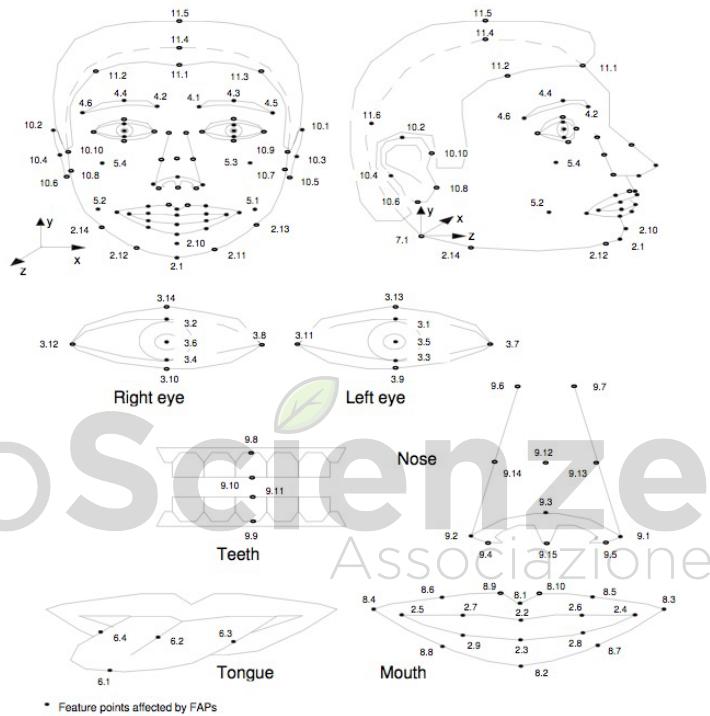


Figure 2.3 MPEG-4 feature points

## Problemi

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 piuttosto piccoli, e tendono progressivamente a determinare effetti che col tempo possono rendere totalmente inefficaci i processi di riconoscimento.

## Verifica e riconoscimento

- **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.

### 7.1.1 Struttura di un riconoscitore facciale

Il primo passo è legato ad individuare all'interno di un video/immagine la presenza di un volto (Face detector). 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 (Face Segmentation). Il primo problema da risolvere, è quello di costruire un face detector efficace, ovvero come capire se in un'immagine vi è un volto.

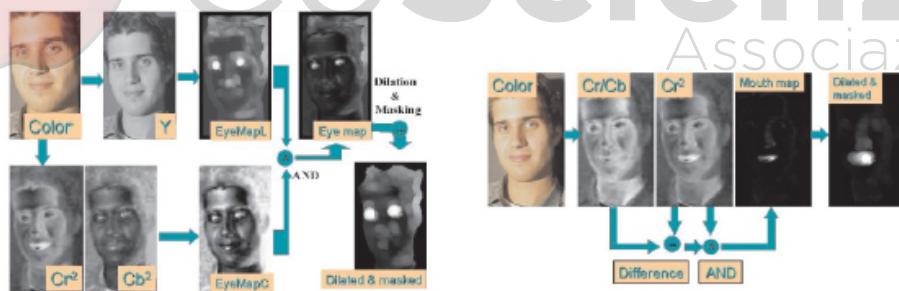
### 7.1.2 Face localization

Le tecniche per effettuare operazioni di Face Detection sono diverse. Le più elementari ma comunque estremamente performanti sono quelle basate su caratteristiche di basso livello che possono riguardare alcune caratteristiche dei pixel (luminosità del volto 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. Tutte queste sono caratteristiche di basso livello nelle quali si possono individuare le tecniche di face detection **features-based**; esse sono tecniche che sfruttano alcune proprietà di basso livello del volto per individuarlo all'intero di una immagine.

Esistono anche approcci che piuttosto che basarsi sulle caratteristiche elementari del volto che 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 e sia immagini non contenenti volti, e si allena il sistema a riconoscere soltanto i volti.

### 7.1.3 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 una 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 C C (crominanza). Nella componente di crominanza sono particolarmente evidenti 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).



### 7.1.4 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.

## 7.2 Face Detection: Viola-Jones

Proposto da Paul Viola e Michael Jones è uno degli approcci di maggior successo (finora) per la 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 e contiene tali informazioni:

- **Misses:** un oggetto presente non viene rilevato;
- **False Alarms:** un oggetto viene rilevato ma non è presente.

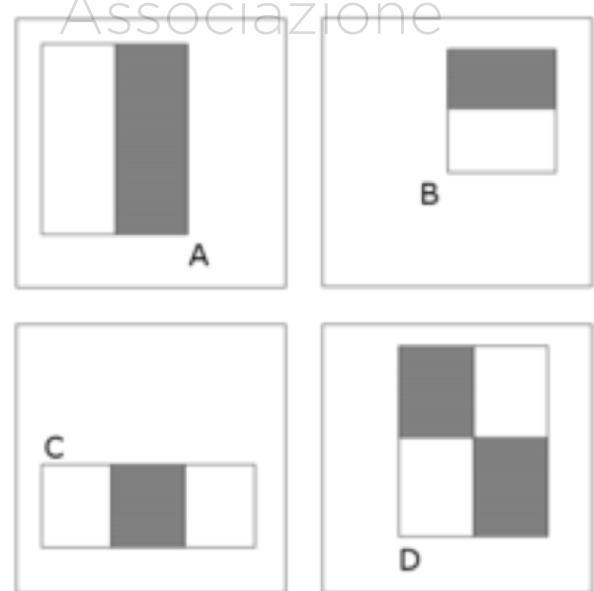
Possono essere ridotti riqualificando l'aggiunta di nuovi esempi adatti (positivi o negativi).

L'algoritmo di **Viola-Jones** è un algoritmo composito in cui ci sono alcuni aspetti che di fatto mettono insieme un pò 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. 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 sovrapposti 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 molti 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 diversi 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 supera 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. 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 rettangoli ombreggiati. Caratteristica rettangolo: **Valore =  $\Sigma$  (pixel nell'area nera) -  $\Sigma$  (pixel nell'area bianca)**.



Per individuare un volto in un'immagine quindi quello che si fa è creare una maschera ed iniziare ad applicare tutti i filtri facendola scorrere. 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 (succede quando un determinato numero di filtri ha dato esito positivo).

Il problema è che ogni volta viene applicata una maschera 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**.

### 7.2.1 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:

$$II(x_0, y_0) = \sum_{x=0}^{x_0} \sum_{y=0}^{y_0} I(x, y) \quad (7.1)$$

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 solamente 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 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.

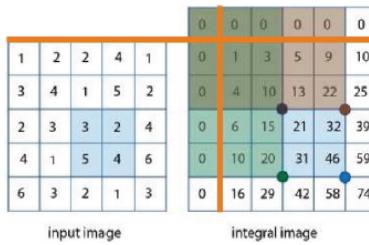


Figura 7.1: Caption

### 7.2.2 AdaBoost

Nel 1995 è stato introdotto un nuovo metodo di estrazione delle Haar feature chiamato AdaBoost. 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.

AdaBoost è un algoritmo di apprendimento automatico supervisionato che viene utilizzato per migliorare le prestazioni dei classificatori deboli combinando i loro risultati per creare un classificatore forte. L'algoritmo addestra una sequenza di classificatori deboli, ciascuno dei quali è addestrato su un sottoinsieme del set di dati di addestramento.

Nel contesto della face detection, l'algoritmo AdaBoost può essere utilizzato per migliorare le prestazioni di un classificatore di base, ad esempio un classificatore basato su una cascata di Haar-like features, che è in grado di individuare alcune caratteristiche del volto, come gli occhi, il naso e la bocca. Il classificatore AdaBoost viene addestrato su un grande set di immagini di volti e non volti, in modo da apprendere le caratteristiche che distinguono i volti dagli altri oggetti presenti nell'immagine. In ogni iterazione di boosting:

- Su ogni esempio valuta ogni rettangolo che fa da filtro;
- Per ogni filtro seleziona la migliore soglia;
- Seleziona la migliore combinazione filtro/soglia;
- Riassegna i pesi sugli esempi (Reweight examples).

La complessità computazionale del learning è : **O(MNT)** dove M sono il numero di filtri, N il numero di esempi e T la thresholds (soglia).

## 7.3 Face Recognition

Una volta effettuata l'operazione di detection, si procede all'estrazione di questo volto dall'immagine, precisamente le 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. 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);
- **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.

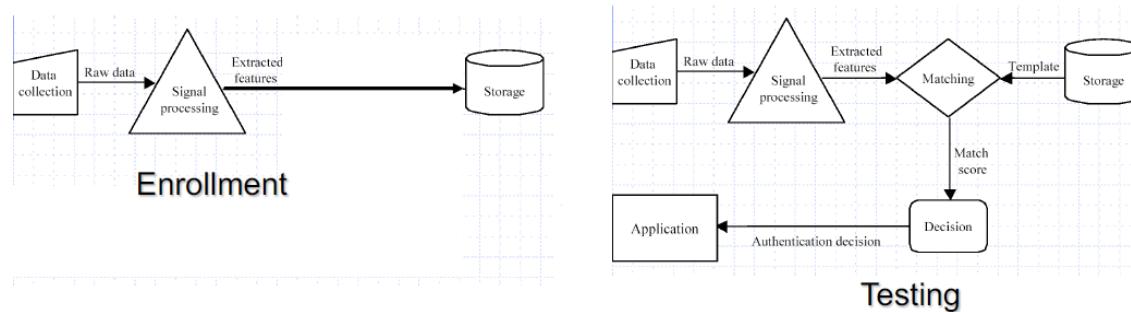
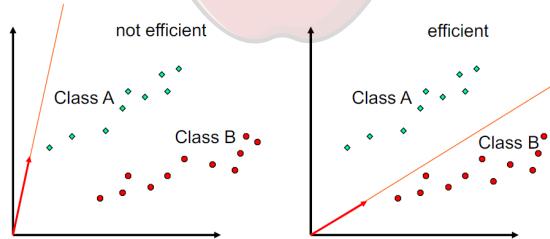


Figura 7.2: Caption

### 7.3.1 ICA e PCA

Tra le tecniche più utilizzate soprattutto in passato per il Face Recognition c'è la **PCA (Principal Component Analysis)** che rientra nella categoria delle Image Based. Tale tecnica sotto alcuni aspetti ricorda la trasformata di Fourier, ma con un approccio totalmente diverso. Piuttosto che considerare i pixel, si associa un'immagine ad un vettore (direzioni) in cui è massima la variazione (varianza) tra i vettori stessi.



L'obiettivo principale della PCA è di individuare dei vettori che siano efficienti per rappresentare determinati dati. Si veda la figura, ad esempio 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 assuma 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.

### 7.3.2 PCA: il metodo

1. «Training set» costituito da un insieme di immagini delle stesse dimensioni e normalizzate in maniera da evidenziare caratteristiche del volto, quali occhi e bocca;
2. 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 decompon la struttura di un volto in una combinazione di componenti ortogonali, correlate tra loro, dette **eigenface**;
3. 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 un volto con le altre presenti nel training set viene effettuato semplicemente valutando la distanza tra questi vettori di caratteristiche locali;
4. La matrice di covarianza: è la matrice simmetrica che rappresenta la variazione di ogni variabile rispetto alle altre (inclusa se stessa):
  - Se valori grandi (rispetto al valore atteso) di X si accostano a valori grandi di Y, oppure valori piccoli di X con valori piccoli di Y, allora la covarianza è maggiore o uguale a 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, allora la covarianza sarà minore o uguale a 0 e le variabili aleatorie in questione si diranno **correlate negativamente**.

Ridurre la dimensione implica perdere informazione. PCA cerca di preservare più informazione possibile minimizzando l'errore tramite la normalizzazione.

### 7.3.3 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 i 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. Una volta linearizzati i volti prima di procedere a risolvere il sistema lineare, si calcola un valore  $\psi$  cioè il valore medio di tutti i volti, si procede poi per ogni vettore a togliere la media. Questo viene fatto per eliminare tutti le informazioni comuni tra i volti e rendere così evidenti le parti più significative. Fatto questo si procede a risolvere il sistema ed a ottenere i vettori indipendenti. Si rappresenta poi ciascun volto come rappresentazione di una combinazione dei volti dei vettori indipendenti. I coefficienti utilizzati per la rappresentazione costituiscono il vettore di caratteristiche.

Abbiamo due tipi di covarianza:

- **Covarianza positiva:** indica che due variabili tendono a muoversi nella stessa direzione;
- **Covarianza negativa:** rivela che due variabili tendono a muoversi in direzioni inverse.

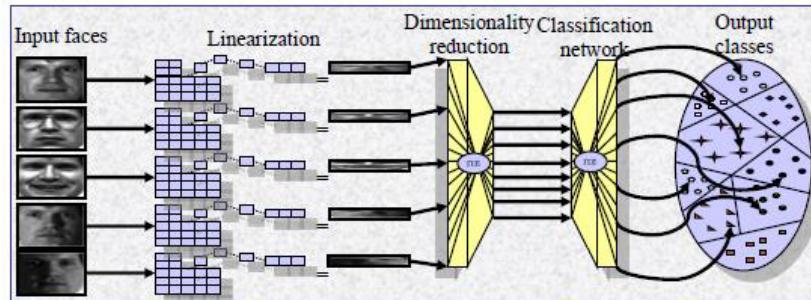
**PCA ed Eigenfaces** : Riassumendo, il principio alla base della PCA è la ricerca all'interno della base di dati delle immagini meno correlate aspettandosi che esse siano dal punto di vista numerico molto inferiore alla dimensione dell'intero database. Quindi, se il database possiede M immagini, si 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 e svantaggi

- **Vantaggi:** La fase di identificazione è veloce. Se si conservano gli autovettori 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, ...

### 7.3.4 Reti neurali

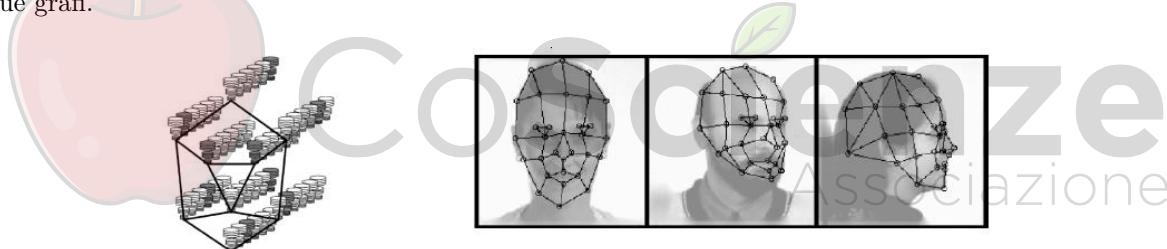
Una rete neurale simula il funzionamento dei neuroni nel cervello. Ciascun neurone è rappresentato da una funzione matematica basata sul calcolo delle probabilità. Per il riconoscimento dei volti, l'ottimo sarebbe utilizzare un neurone per ciascun pixel. Questo approccio richiede troppi neuroni. La soluzione 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.

### 7.3.5 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 re-training del sistema.
- **Svantaggi:** il processo di testing è molto lento perché richiede il confronto fra grafi (NP-Hard).

### 7.3.6 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.

### 7.3.7 Perché utilizzare il volto allora?

Uno dei motivi del perché utilizzare il volto è dovuto alla sua **università**. Abbiamo anche un **efficiente** uso di memoria (chiave di 84 bytes o 2k). È molto efficace perché è facile da verificare. Alta **misurabilità** accettazione. L'**acquisizione** può essere attiva o passiva e i costi dei dispositivi non sono alti. Dall'altro canto abbiamo anche molti svantaggi, ossia non è **unica** (sosia), non è sempre **efficace** nel riconoscimento, poco **stabile**, c'è molta **variabilità intra-classe** (umore, smorfie, ecc). Infine è molto facile da **raggirare**. Un modo per aggirare la face detection è indossare indumenti con particolari trame che durante la fase di detection portino a creare più finestre sovrapposte facendo così tutti i volti trovati siano scartati.

# Content Based Image Retrieval (CBIR)

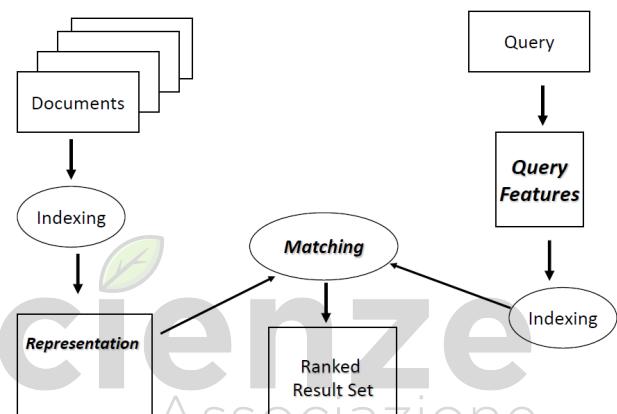
Sono sistemi che si sono sviluppati con l'aumento della mole di immagini e video oltre il quale una indicizzazione non è più fattibile. Si tratta di sistemi dove il recupero delle immagini viene effettuato sulla base del contenuto. Quindi, un CBIR, è sistema il cui obiettivo è quello di consentire il recupero di immagini da un database che siano rilevanti rispetto ad una query; il sistema è in grado di fornire come risposta un insieme d'immagini che possiedono delle caratteristiche di similarità con l'immagine che si è utilizzata come query o con la query stessa. Quando parliamo di similarità bisogna capire se si fa riferimento all'intera immagine o ad una parte dell'immagine stessa. Trovare una immagine che sia simile ad una query è concetto astratto che non sempre si riesce a codificare precisamente.

Le immagini vengono ordinate e disposte mediante analogie di similarità molto forti tra loro. I CBIR realizzano i **match inesatti**, cioè la risposta è individuabile in un insieme di immagini ordinate per similarità dove l'utente effettua una successiva scelta. Quindi a fare la differenza sul concetto di similarità, spesso oltre ad essere l'utente è l'ambito applicativo in cui si opera (es: distribuzione dei colori in una parte di essa).

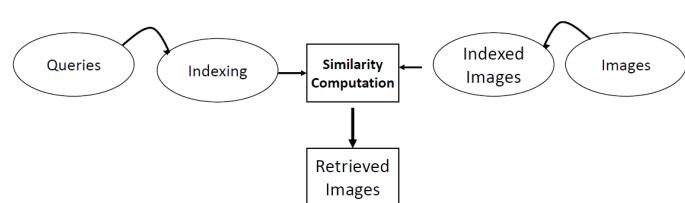
## Interrogazione e recupero in un CBIR

Il processo di indicizzazione basato su contenuto è molto più complesso di un sistema **text retrieval** (il sistema recupera solo il file con quel dato nome), in quanto va specificata la realizzazione di un query, successivamente come indicizzare e rappresentare un oggetto multimediale e infine come costruire l'architettura di un CBIR.

Il classico sistema di information retrieval è un sistema abbastanza schematico, dove ci sono un insieme di documenti dove ciascuno di essi viene opportunamente indicizzato e rappresentato sotto forma di metadati all'interno di un database. L'indice viene associato al documento e ne rappresenta il suo metadato rispetto al quale vengono effettuate le ricerche. Durante la fase di query, viene posta appunto la query: in questo caso l'oggetto è stato indicizzato in forma testuale (la query è testuale) l'indice rappresentato attraverso le tecniche di matching viene individuato nella sua rappresentazione e viene fornita poi la risposta che può essere composta di una singola immagine (match esatto) o una serie di immagini (match inesatto).



Il concetto di CBIR è analogo al precedente, l'unica cosa che varia è l'uso di un match inesatto. Infatti, non si vuol trovare un'unica immagine, ma si cerca un concetto di similitudine, ovvero un insieme di risposte nel quale poter navigare e cercare poi le immagini che sono più pertinenti rispetto al contesto.



Nel contesto della **Text retrieval** si sa che per indicizzare una foto/video, il testo è fondamentalmente oggettivo (si associa alla parte testuale un significato semantico). Il concetto di immagine è più complesso, innanzitutto perché i pixel non hanno un significato semantico in quanto ad un livello di luminosità non è possibile associare un significato, in quanto esso dipende dal contesto dell'immagine, e non dal livello di luminosità. Nell'Image Retrieval è possibile essere interessati a un sottoinsieme dell'immagine, quindi bisogna effettuare una segmentazione di tali oggetti per poterli descrivere (operazione abbastanza complessa). Infine, la descrizione di un oggetto visuale dipende da molti fattori come il punto di vista, l'illuminazione ecc..., un oggetto può anche cambiare dal punto di vista visuale a seconda del contesto in cui è individuato.

## Similarità

La prima cosa che deve essere definita quando si utilizza un CBIR è il concetto di similarità, in quanto esso permette di concentrare l'attenzione sulle caratteristiche da estrarre da un'immagine. Nel caso delle auto il concetto di similarità potrebbe essere il modello, il colore dell'auto, ecc ... Tutto ciò ha un'importanza in quanto la definizione di similarità è legata alla definizione delle caratteristiche che si vogliono estrarre, o al concetto di metrica per misurare se due immagini sono distanti o simili, e all'individuazione delle **non-metric distance**, quindi misure di similarità che non si basano su metriche tradizionali. Il primo elemento considerato quando si introduce il concetto di similarità è che le misure di similarità introducono due elementi di disturbo:

- **False alarms:** immagini non significative inserite nell'insieme risposta (answer set).
- **False Dismissals:** immagini significative non inserite nell'insieme risposta, ritenuto anche il più grave.

Quindi la progettazione di un CBIR deve essere fatta in modo tale da limitare i due problemi appena elencati in particolare i False Dismissals, perché se un'immagine non viene recuperata, essa può tramutarsi in un problema serio. Nel caso di un False Alarms si può scartare l'immagine non pertinente al contesto.

## 8.1 Image Features

L'immagine è un concetto piuttosto complesso il quale può essere interpretato a vari livelli e ciò dipende da vari fattori. Nell'immagine è possibile immaginare tre livelli di caratteristiche:

- **Caratteristiche primitive (basso livello):** colori, forme, relazioni spaziali, ecc...
- **Caratteristiche derivate o logiche:** dove si cerca un certo oggetto all'interno dell'immagine come un volto, o un oggetto di certa forma ecc...
- **Caratteristiche astratte:** che sono in grado di descrivere aspetti astratti come eventi, significati emotivi ecc...

Ovviamente più si passa dalle caratteristiche primitive a quelle astratte, più è complesso il processo di indicizzazione e quindi d'estrazione delle caratteristiche.

### 8.1.1 Image features: primitive

Descrizione delle immagini sotto forma di caratteristiche primitive. Per esempio si possono utilizzare **query elementari** basate sulla ricerca per colore e posizione di un determinato oggetto, oppure tutte le immagini che contengono un elemento all'interno di un altro. O più comunemente si utilizzano delle query dove viene richiesta la ricerca di immagini simili all'immagine query fornita; quindi si parte da un'immagine e si chiede al sistema di recuperare tutte le immagini simili rispetto all'immagine iniziale.

L'utilizzo dei colori o la distribuzione di essi in alcuni casi può produrre dei risultati significativi e in altre occasioni no. Quindi le caratteristiche primitive, in particolare i colori, possono essere utilizzati per discriminare immagini molto diverse tra loro; avendo un database di immagini eterogenee provenienti da vari ambiti applicativi o contrasti sociali, le caratteristiche primitive come il colore possono essere utilizzate per classificare immagini simili, con la forte probabilità di introdurre molti False Alarms.

Un'indicizzazione secondo i colori può essere effettuata tramite un istogramma dove abbiamo sull'asse delle ascisse tutti i possibili colori e sull'asse delle ordinate le occorrenze di ciascuno di essi, ottenendo un vettore n-dimensionale che rappresenterà la chiave di indicizzazione dell'immagine; tale vettore presenta per ogni elemento un colore e il numero associato a quella posizione risulta essere la percentuale con cui quel colore viene rappresentato all'interno dell'immagine.

#### Algoritmo di match dell'istogramma

Una volta individuate le caratteristiche e la loro rappresentazione, bisogna individuare una modalità per effettuare il match. È possibile utilizzare una metrica euclidea dove viene calcolata una distanza tra le immagini, rappresentate come vettori, nello spazio n-dimensionale. Questo però consta di un tempo computazionale elevato data la mole di colori da dover confrontare ( $2^{24}$  in alcuni casi); il tempo di risposta che si cerca deve essere molto basso.

#### Metrica euclidea

- spazio n dimensionale:
  - Image 1:  $(C_{11}, C_{12}, \dots, C_{1n})$
  - Image 2:  $(C_{21}, C_{22}, \dots, C_{2n})$

le n-componenti sono alla fine l'insieme di risposte migliori possibili utilizzando il colore.

Si può migliorare questo processo combinando la trasformata di Fourier e la metrica euclidea, prendendo il vettore dell'istogramma, tagliando le frequenze, portando avanti soltanto i primi coefficienti quelli relativi alle frequenze più basse, con k (soglia di taglio) molto più piccolo di n. Quindi piuttosto che operare su tutti gli elementi, si trasforma il vettore nello spazio delle frequenze ottenendo n frequenze e di esse si portano avanti soltanto le prime k. Con un'operazione simile calcolando una differenza su k elementi, la loro distanza nello spazio delle frequenze sicuramente non aumenterà. Questo può portare alla generazione di più false alarms.

## Teorema di Parseval

Se è la trasformata di Fourier di una sequenza discreta  $x$  (come ad esempio un istogramma) si avrà che: la sommatoria degli  $n$  elementi nel dominio spazio-tempo è uguale alla sommatoria degli  $n$  elementi nel dominio delle frequenze, avendo una conservazione dell'energia. Quindi, se si sommano gli  $n$  campioni nel dominio spazio-tempo, tale somma sarà uguale alla sommatoria dei coefficienti delle armoniche nel dominio delle frequenze.

$$\sum_{i=0}^{n-1} x_i^2 = \sum_{u=0}^{n-1} X_u^2 \quad (8.1)$$

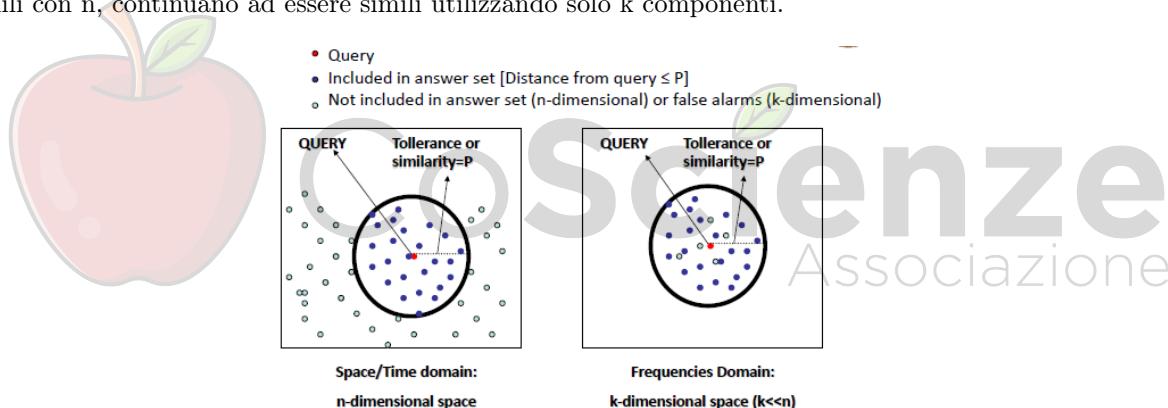
Considerando due vettori (due istogrammi) nel dominio spazio-tempo dove  $D$  è una distanza euclidea o una qualsiasi metrica, la distanza tra  $x$  e  $y$  è uguale alla distanza delle rispettive trasformate di Fourier  $X$  e  $Y$ . Vale con qualsiasi metrica.

$$D(\vec{x}, \vec{y}) = D_n(\vec{X}, \vec{Y}) \quad (8.2)$$

Sapendo che nel dominio delle frequenze cerchiamo di conservare solo le frequenze più basse, se consideriamo una metrica euclidea calcolando la distanza sulle prime  $k$  frequenze, essa è sicuramente minore o uguale della distanza euclidea su  $n$  frequenze, e si avrà come risultato che  $D_k \leq D_n$ . Quindi il teorema di Parseval permette di stabilire fondamentalmente che, introducendo questo passaggio, non si avranno False Dismissals. Ciò significa che se si fosse stato in presenza di immagini simili utilizzando tutte le componenti dell'istogramma, utilizzandone soltanto  $k$  la distanza non aumenta e quindi se l'immagine è pertinente ( $x$  simile ad  $y$ ) lo sarà ancora quando si utilizzano solo  $k$  componenti. Scegliendo un  $k$  molto piccolo, o comunque più piccolo di  $n$ , non vengono introdotti False Dismissals rispetto alla ricerca effettuata considerando tutte le  $n$  componenti. Il concetto di False Dismissals è legato all'utilizzo di tutte le componenti.

$$D_k(\vec{X}, \vec{Y}) \leq D_n(\vec{X}, \vec{Y}) \quad (8.3)$$

Se il sistema, utilizzando l'istogramma dei colori introduceva dei False Dismissals, lo introdurrà anche utilizzandone  $k$ , ma l'idea è di porsi nel caso in cui si considerano tutte le  $n$  componenti, si cerca una strategia per non prendere tutte le componenti ma soltanto  $k$  e dimostrare che pur prendendone  $k$  si ottiene lo stesso risultato che si otteneva prendendo tutte le  $n$ . Quindi due immagini simili con  $n$ , continuano ad essere simili utilizzando solo  $k$  componenti.



Facendo questa operazione è possibile introdurre dei False Alarms, in figura una simulazione nello spazio bidimensionale. Quando viene diminuita la dimensionalità dello spazio, le immagini che in precedenza non erano significative che si trovavano all'esterno del cerchio rappresentato, ora diventano significative, e tutte le similarità che erano presenti in precedenza continuano ad esserci, ma abbassando la dimensionalità potranno entrare nel cerchio alcune immagini che prima erano al di fuori di esso. Infatti, considerando una dimensionalità più bassa  $k$ , alcuni di quelli che su  $n$  sarebbero stati esclusi, adesso con  $k$  più piccolo di  $n$  potrebbero rientrare, rappresentando i False Alarms. Applicando il teorema di Parseval, passando per Fourier e riducendo la dimensione non si avranno dei False Dismissals ma si potrebbero introdurre dei False Alarms.

Attraverso la trasformata di Fourier si ottiene un ulteriore elemento di applicazione per ridurre la dimensionalità di spazi particolarmente ampi, senza compromettere il risultato finale che si sta cercando. Altre applicazioni del teorema di Parseval combinato con la trasformata di Fourier, possono essere trovate nel caso in cui si voglia rappresentare la forma di un oggetto. Essa può essere rappresentata considerando il baricentro dell'oggetto e considerando le distanze di ciascun punto del contorno dal baricentro, ottenendo un segnale monodimensionale. La forma del contorno attraverso la rappresentazione monodimensionale è composta da un gran numero di punti. Su questo segnale è possibile applicare la trasformata di Fourier e prendere in considerazione soltanto  $k$  componenti con  $k$  molto piccolo, avendo una rappresentazione del contorno senza introdurre False Dismissals. Un'altra possibilità è quella di poter rappresentare una tessitura con analogo concetto.

### 8.1.2 Image features: logical

Le caratteristiche logiche risultano esse più complesse da estrarre da un'immagine, infatti esse richiedono tecniche di computer vision, spesso realizzate attraverso una conoscenza specifica del dominio di riferimento. I CBIR utilizzano poco questo tipo di caratteristiche in quanto risultano essere complesse e computazionalmente onerose da implementare.

### 8.1.3 Image features: abstract

Ancor più complessa risulta essere la fase in cui le immagini vengono indicizzate con caratteristiche astratte. Si passa quindi ad attributi visuali che descrivono situazioni particolari come ad esempio contesti emotivi, culturali ecc.. e richiedono un processo di elaborazione dell'immagine piuttosto complesso.

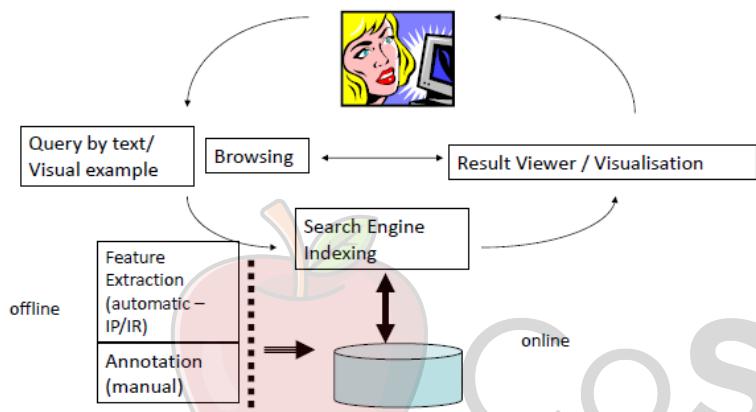
- recupero di eventi o tipi di attività nominati (ad es. "trova immagini di danze popolari scozzesi");
- recupero di immagini con significato emotivo o religioso ("trova a quadro raffigurante la sofferenza").

#### Livelli di descrizione delle immagini

- **metadati indipendenti dal contenuto:** dati che non riguardano direttamente il contenuto di un'immagine, ma in qualche modo correlato ad esso (es: nome di un autore, data, luogo, ecc..);
- **Dati che si riferiscono al contenuto visivo di immagini:** caratteristiche intermedie basse (colore, tessitura, forma ecc.) noti come metadati dipendenti dal contenuto, i dati si riferiscono alla semantica del contenuto.

**Attributi delle immagini :** Sono attributi di un'immagine il colore, la texture, l'aspetto e la shape, e spesso sono correlate con la semantica (es: recuperare le foto di Bill Clinton in mezzo alla folla usando la somiglianza per aspetto).

## 8.2 Architecture of Visual Information retrieval system



In figura viene mostrata l'architettura di un sistema CBIR di nuova generazione. Concettualmente non è diversa da un sistema tradizionale di archiviazione, con l'unica differenza che la query può essere posta o sotto forma di esempio o sotto forma di immagine. Fornendo al sistema una immagine, ed esso sarà in grado di estrarre determinate caratteristiche dall'immagine e di ricercare all'interno del database immagini le cui caratteristiche sono simili a quelle poste dall'immagine iniziale. Quindi il processo di annotazione visuale risulta essere automatico e non supervisionato dall'utente. Molti sistemi combinano i due aspetti sia automatico che manuale.

**NOTA:** è molto più facile costruire un sistema semiautomatico rispetto a un sistema completamente automatico. Perché si assume che una persona utilizzerà il sistema e sarà in grado di fornire un feedback.

**Query paradigms** I paradigmi che vengono utilizzati sui CBIR possono essere due:

- **Query by Example (QbE):** l'utente fornisce un'immagine come query;
- **Query by Sketch (QbS):** l'utente costruisce lo sketch dell'immagine.

#### Indicizzazione manuale vs CBIR

- **Indicizzazione manuale:** possibilità di utilizzare una vasta gamma di software di recupero testuale per accedere a informazioni e conoscenze specifiche relative alle immagini presenti in tali librerie. Ma il lavoro da fare è molto intenso, può essere soggettivo e inaffidabile, è difficile catturare il concetto di somiglianza;
- **CBIR:** Il feature matching è oggettivo e automatico, le query sono formulate tramite un processo visivo. Il conto è che non è possibile capire la semantica del contenuto di un'immagine catturata, le feature non matchano precisamente con il concetto di similarità di un essere umano.

## 8.3 Misure per un CBIR

L'efficienza di un CBIR viene misurata in vari modi, dove ogni misura dipende spesso dal dominio di applicazione:

- **Efficienza:** si misura con il tempo di risposta medio per effettuare un recupero;
- **Efficacia:** riduzione della quantità di false Alarms e false Dismissals;
- **Minimizzare False Alarms;**
- **Minimizzare False Dismissals.**

Per quanto riguarda i False Alarms e i False Dismissals esistono due misure che sono:

- **Recall:** capacità del sistema nel recuperare tutte le immagini significative;
- **Precision:** la capacità del sistema nel recuperare solo le immagini significative.

Queste due misure vengono calcolate nel seguente modo: si supponga di avere una collezione di immagini (database) con **TOT** elementi, e per ogni possibile query (supervisione di un esperto) si associa a ciascuna delle immagini un Rango che va da 1 a **REL**, dove **REL** rappresenta, per ogni possibile query che può esser effettuata, il grado di similarità. Infatti, se rispetto ad una query si associa ad un'immagine un rango 1, essa è più simile alla query rispetto ad un'immagine con rango 2 ecc.., più cresce il rango più l'immagine risulta essere difforme in termini di similitudine dall'immagine query.

Per ogni query si avranno un certo numero di immagini rilevanti. **IR (Ideal Rank)** dovrebbe essere un valore molto piccolo:

$$IR = \sum_{r=1}^{REL} \frac{r}{REL} \quad (8.4)$$

**AR (average rank)** viene spesso utilizzato come metrica per misurare quanto bene il sistema recupera le immagini rilevanti. Un average rank più basso indica che le immagini rilevanti sono posizionate più in alto nei risultati di ricerca, il che generalmente è considerato una migliore prestazione.

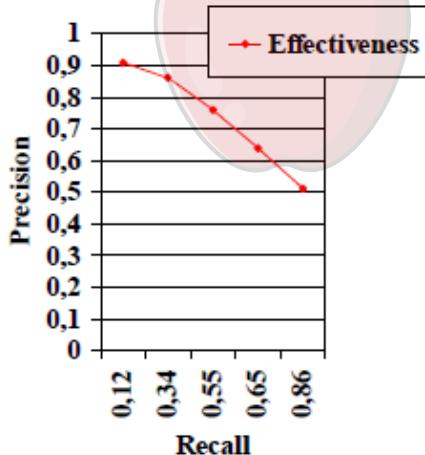
$$AR = \sum_{r=1}^{REL} \frac{rank_r}{REL} \quad (8.5)$$

La **Normalized Recall (NR)** è una misura efficace per misurare il comportamento di un CBIR. È possibile misurare CBIR differenti e controllare mediamente quali sono i valori di NR prodotti.

- **Effettiva = (AT - IR)**, il risultato è nel range [0; (TOT-REL)] dove:
  - 0 (**AR-IR**) è il recupero perfetto;
  - (**TOT-REL**) è il caso peggiore.
- **Normalized Recall (NR)** il risultato è nel range [0,1] dove 1 è il recupero perfetto e 0 il caso peggiore.

$$NR = 1 - \frac{(AR - IR)}{(TOT - REL)} \quad (8.6)$$

### 8.3.1 Recall vs Precision



Quando si interroga un CBIR bisogna introdurre una tolleranza, ovvero un errore che si vuol commettere. Per capire se il sistema è in grado di recuperare tutte le immagini significative, bisogna aumentare progressivamente l'errore. Aumentando l'errore, la capacità del sistema nel recupero tutte le immagini significative, ovviamente aumenta, e aumentano anche il numero di immagini significative che si inseriscono. Aumentando l'errore, inoltre, succede che viene inserita anche informazione inutile. Ciò fa in modo che la precisione tende a diminuire. Quindi con valori di Recall piccoli, la precisione è molto alta, infatti si introducono pochi False Alarms. Ma quando la Recall tende ad aumentare, siccome si aumenta la tolleranza, la Recall cresce in quanto più abile nel recuperare tutte le immagini ma la Precision ne risente e diminuisce. Questo rapporto permette di stabilire l'efficacia di un sistema CBIR. Un sistema CBIR ideale dovrebbe mantenere una Precision piuttosto alta, man mano che la curva tende a scendere verranno introdotti molti False Alarms.

# Noisy Iris Detection and Recognition

**Perché l'iride?** 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 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  $3 \text{ cm}^2$ . 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 ecc...

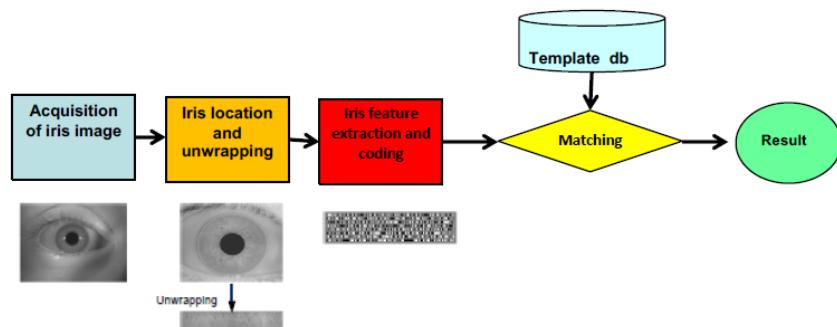
L'iride come accennato può essere acquisita sia nel visibile che nell'infrarosso. La differenza sostanziale è legata alla quantità di melanina. Ovvero nel caso della luce visibile, la melanina assorbe la luce visibile, evidenziando gli strati che compongono l'iride sono visibili. L'immagine ottenuta con la luce visibile non permette di estrarre tutte le informazioni tridimensionali per una serie di problematiche. 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 si possono evidenziare 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, o un'immagine con Blurring, un'occlusione oppure 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.

## Fase di preprocessing

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

1. **Segmentazione (o detection):** quindi individuare che all'interno di un'immagine è presente un iride;
2. **Normalizzazione:** l'iride estratta può essere variabile per via di contaminazioni. Anche la dimensione delle pupille può portare problemi. La normalizzazione serve a uniformare l'iride così da poterlo lavorare;
3. **Coding:** che permette di estrarre le caratteristiche sulla base di un modello diversificato da caso a caso;
4. **Matching:** dove si confronta il codice prodotto nella fasi precedenti.



## 9.1 Algoritmo di 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. 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. Il problema principale come detto è riuscire ad individuare l'iride. Se non si inizia tale processo o lo si sbaglia, tutto ciò che viene dopo risulta essere fortemente compromesso.

### 9.1.1 Daugman: iris location

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 e raggio variabile.

$$G_\sigma(r) = (1/\sqrt{2\pi\sigma}) \exp[-(r - r_0)^2/2\sigma^2] \quad (9.1)$$

Il filtro gaussiano era importante per determinare i cerchi, infatti lo smoothing serviva a evitare che le 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 veniva 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 **integrale curvilineo** che infine determina i candidati ideali a identificare l'iride.

$$\max_{(r,x_0,y_0)} |G_\sigma(r) * \frac{\alpha}{\alpha r} \int_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds| \quad (9.2)$$

**NOTA:** visto che l'algoritmo di Daugman lavora sulla creazione di circonferenza la presenza di rumore o luce riflessa può portare alla creazione di circonferenze in zone errate e quindi ad una errata detection.

### General Iris Segmentation

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.



Figura 9.1: Daugman

### 9.1.2 Daugman : iris unwrapping

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 normalizzata, proprio perché le aree utili erano diverse. Daugman applicò quindi vari algoritmi combinati tra loro che attraverso la combinazione tra coordinate

polori 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.

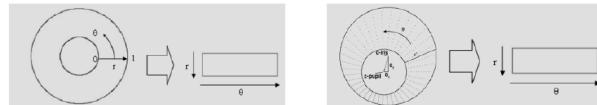


Figura 9.2: Daugman

### 9.1.3 Daugman : feature extraction

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.

## 9.2 NICE (Noisy Iris Challenge Evaluation)

NICE (Noisy Iris Challenge Evaluation) è un evento di valutazione che si concentra sull'analisi delle iridi con immagini oculari di scarsa qualità o affette da rumore. L'obiettivo principale di NICE è sfidare gli algoritmi di riconoscimento dell'iride a operare in condizioni difficili, simili a quelle che potrebbero verificarsi nella realtà, come immagini con scarsa illuminazione, offuscamento, occlusione parziale o altri fattori di degrado. È composto da due fasi:

- NICE.I : è la prima fase dell'evento, in cui i partecipanti sono sfidati a sviluppare e valutare i loro algoritmi di riconoscimento dell'iride utilizzando un insieme di dati specifico contenente immagini oculari con rumore o scarsa qualità. Durante questa fase, gli algoritmi vengono testati e i risultati vengono analizzati per valutare le prestazioni in condizioni di degrado delle immagini;
- NICE.II : rappresenta la seconda fase dell'evento, in cui vengono introdotti nuovi set di dati contenenti immagini oculari con ulteriori sfide e condizioni di rumore. I partecipanti sono nuovamente sfidati a sviluppare e valutare i loro algoritmi utilizzando questi nuovi dati per testare le prestazioni dei loro metodi di riconoscimento dell'iride in scenari più complessi.

### 9.2.1 NICE.I

Il database UBIRIS sono stati creati dal SOCIA Lab. presso l'Università di Beira Interior in Portogallo. Questi database contengono immagini dell'iride catturate in condizioni di illuminazione diverse, che hanno causato la comparsa di immagini molto degradate. Le immagini dell'iride sono state acquisite sia con l'illuminazione naturale che artificiale. Questi database rappresentano una sfida per gli algoritmi di riconoscimento dell'iride a causa delle condizioni di illuminazione eterogenee e delle immagini altamente degradate. I volontari erano per il 90% caucasici, 8% negri (mi dissocio), 4% asiatici. Il 60% dei volontari hanno partecipato in due sessioni, il 40% in una sola. Le acquisizioni avvenivano guardando in diversi target su di un percorso da diverse angolature. Tra le due sessioni luce, posizione dei device e la loro orientazione. Per capire l'algoritmo vincitore si effettuavano degli XOR tra le varie immagini in output di ogni algoritmo con quelle scontornate a mano.

L'algoritmo vincitore di questa prima sfida fu il:

**"Casia Algorithm by T. Tan, Z. He and Z. Sun (the Chinese Academy of Sciences)"**

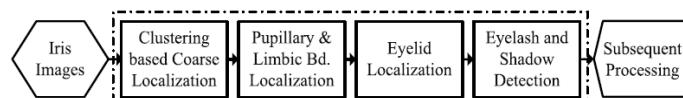


Figura 9.3: Daugman

Quest'ultimi furono i vincitori anche della seconda fase.

# Fingerprint

Le impronte digitali sono un'identificazione biometrica unica e personale. Sono importanti perché sono utilizzate per l'identificazione personale, la sicurezza e l'investigazione forense. Le autorità le utilizzano per collegare persone a determinati crimini, mentre nel campo della sicurezza vengono usate per il controllo degli accessi. Tuttavia, la digitalizzazione delle impronte digitali presenta sfide, come la necessità di una scansione accurata e l'implementazione di misure di sicurezza per proteggere i dati sensibili. Nonostante queste difficoltà, le impronte digitali rimangono uno strumento essenziale grazie alla loro unicità e affidabilità.

## Digitalizzazione delle impronte digitali

I principali parametri che caratterizzano l'immagine digitale delle impronte sono:

- la risoluzione: determina la nitidezza e la precisione dei dettagli dell'immagine delle impronte digitali;
- l'area di acquisizione: rappresenta la dimensione fisica dell'area delle dita che viene scansionata per ottenere l'immagine delle impronte digitali;
- il numero di pixels: indica la quantità di informazioni presenti nell'immagine delle impronte digitali, influenzando la sua qualità e dettaglio;
- l'accuratezza geometrica: si riferisce alla fedeltà della rappresentazione dell'immagine rispetto alla morfologia reale delle impronte digitali;
- il contrasto: riguarda la differenza di intensità tra le aree chiare e scure nell'immagine delle impronte digitali, facilita l'analisi delle caratteristiche distintive;
- la distorsione geometrica: rappresenta qualsiasi deformazione o alterazione dell'immagine che può influire sulla sua qualità e accuratezza.

## Tecnologie per la Digitalizzazione delle Impronte Digitali

Tecnologie per la Digitalizzazione delle impronte digitali:

- **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 micro-variazioni 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 lunghezze d'onda visibili ma nella gamma dell'infrarosso.

## Caratteristiche di uno Scanner di impronte Digitali

Le principali caratteristiche di uno scanner d'impronti digitali dipendono dal sensore utilizzato e influenzano le caratteristiche dell'immagine risultante. Queste caratteristiche includono:

- **Dpi (Dot per inch)**: Misura la risoluzione di scansione in termini di densità di punti per unità di misura.
- **Area utile di acquisizione**: Le dimensioni della superficie sensibile influenzano il numero di caratteristiche distintive acquisite e l'accuratezza.
- **Gamma dinamica**: Il numero di livelli di grigio quantizzati dallo scanner, che determina la precisione nella rappresentazione dei dettagli.
- **Fps (frames per second)**: Il numero di immagini acquisite e inviate all'host in un secondo. Un frame rate elevato permette una maggiore tolleranza ai movimenti del dito durante l'acquisizione.
- **Rilevamento automatico della presenza del dito**: Alcuni scanner possono rilevare automaticamente la presenza del dito sul sensore, facilitando l'acquisizione.
- **Test di vitalità**: La rilevazione di parametri vitali come il battito cardiaco, la pressione arteriosa e la temperatura.
- **Crittografia**: Alcuni scanner di impronte digitali sono dotati di algoritmi crittografici per proteggere il canale di comunicazione con l'host.
- **Sistemi operativi supportati**: La compatibilità con diversi sistemi operativi, inclusi quelli open-source, può essere importante nella scelta dello scanner in base all'applicazione e all'infrastruttura informatica.

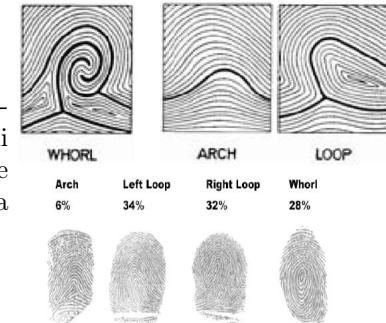
## 10.1 L'impronta digitale

Un'impronta digitale è costituita 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 minuzia (·), che sono punti in cui la struttura della cresta cambia.



### 10.1.1 Caratteristiche globali di un'impronta digitale

Le caratteristiche globali di un'impronta digitale sono un insieme di attributi che descrivono le proprietà generali dell'intera impronta digitale, piuttosto che focalizzarsi sui dettagli specifici dei singoli punti o delle linee presenti. Sono utilizzate per ridurre l'insieme di ricerca dei soggetti da analizzare in un match 1 a molti (nel one to one si usano solo le minuzie). Questa riduzione avviene tramite una classificazione (**clustering**) così da creare classi di ricerca.



I pattern base sono:

#### Loop

In un loop, le linee delle impronte digitali si curvano e formano un motivo ad anello o a forma di S. Ci sono due tipi di loop: il loop radiale, in cui le linee si curvano verso il pollice, e il loop ulnare, in cui le linee si curvano verso il mignolo.

Abbiamo due punti focali: il **core**, o il centro del ciclo, e il **delta**. Il Delta è l'area del modello dove c'è una triangolazione o una divisione delle creste. Quando si registrano le impronte digitali, il delta e l'area tra il delta e il nucleo devono essere completamente registrati. I loop sono presenti nel 60-65% delle impronte digitali.

#### Arch

In un archi, le linee delle impronte digitali si estendono da un lato all'altro, curvandosi leggermente verso l'alto, senza formare un core o delta come nei loop. Gli archi possono essere divisi in due categorie: gli archi radiali, in cui le linee si curvano verso il pollice, e gli archi ulnari, in cui le linee si curvano verso il mignolo.

Presenti nel 5% delle impronte digitali. Possono essere di due tipi:

- **Plain arch (arco semplice)**: le linee delle impronte digitali si estendono da un lato all'altro senza formare una curva centrale o una curvatura significativa. L'arcata del plain arch è piatta e rettilinea, senza alcuna inclinazione pronunciata verso l'alto o verso il basso. È considerato uno dei modelli più semplici e meno comuni nelle impronte digitali;
- **Tented arch (arco tendente)**: le linee delle impronte digitali si estendono da un lato all'altro come nell'archetto, ma con una curvatura più accentuata verso l'alto, formando una sorta di piccolo picco o "tenda" nel centro dell'impronta digitale. Questo picco può avere un aspetto più pronunciato rispetto all'arcata piatta del plain arch.

#### Whorl

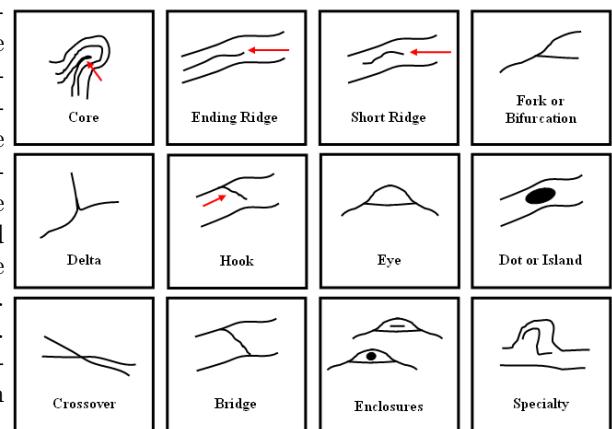
In una spirale, le linee delle impronte digitali formano un motivo a spirale o a vortice. Ci sono diversi tipi di spirali, tra cui la spirale semplice (una sola linea che ruota in un punto centrale) e la spirale doppia (due o più linee che ruotano attorno a due o più punti centrali), in cui ci sono uno o più delta. Le spirali possono essere classificate anche come spirali radiali o ulnari, a seconda della direzione della curvatura delle linee.

Sono presenti nel 30%-35% delle impronte digitali. Abbiamo:

- **Plain whorl (spirale semplice)** e **Central pocket loop whorl (spirale a tasca centrale con loop)**: per capire di quale delle due si tratta basta tracciare una linea tra i due delta del plain whorl e central pocket loop whorl. Se alcune delle creste curve toccano la linea, è una plain whorl. Se nessuno del nucleo centrale tocca la linea, è una Central pocket loop whorl;
- **Accidental Whorl**: contiene 2 o più pattern (escluso l'arco normale) o è un pattern non coperto da altre categorie. Un accidental whorl può consistere di un ciclo di combinazione e semplice spirale o ciclo e arco tenda.

### 10.1.2 Caratteristiche locali di un'impronta digitale (minuzie)

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 locazione 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.



Molto spesso queste minuzie sono situate nelle vallate e non all'interno delle creste dell'impronta digitale. Quindi l'acquisizione o meno delle minuzie dipende da molti fattori come la pressione effettuata dal dito sul sensore, dalla superficie acquisita dal sensore, dall'anzianità del soggetto e dall'oliosità delle dita.

**Nota :** le minuzie sono state scoperte da Dalton.

Quanti dpi? servono per acquisire le varie caratteristiche?

- Globali: 300 dpi;
- Locali (minuzie): tra i 500 e i 1000 dpi;
- Pori: > 1000 dpi



Al livello di osservazione **ultra-fine** è possibile individuare anche dettagli intra-crestae, quali essenzialmente i pori per la sudorazione, la cui posizione e forma sono considerate estremamente distintive. Purtroppo l'estrazione dei pori è possibile solo partendo da immagini delle impronte digitali acquisite ad altissima risoluzione come detto in precedenza, ed in condizioni ideali, per cui questa particolare rappresentazione non è pratica per la maggioranza dei contesti applicativi.

### 10.1.3 Principali approcci metodologici al confronto di impronte digitali

Abbiamo tre principali approcci:

- **confronto basato su correlazione:** le due immagini di impronte vengono sovrapposte e un procedimento di correlazione fra pixels corrispondenti è iterato per i differenti allineamenti ottenuti tramite roto-traslazioni fino a determinare misure di similarità tra i campioni;
- **confronto basato su minutiae:** le minutiae vengono prima estratte dalle due impronte e memorizzate come due insiemi di punti in uno spazio bidimensionale, quindi viene ricercato l'allineamento tra i due insiemi che massimizza il numero di coppie corrispondenti di minutiae e che misura la similarità tra le impronte;
- **confronto basato su caratteristiche delle creste:** poiché l'estrazione delle minutiae in immagini di impronte a bassa qualità è problematico si preferisce utilizzare altre caratteristiche delle creste quali orientamento e frequenza locali, forma delle creste e tessitura in quanto più facili e affidabili da estrarre, ma anche meno distintive. Gli approcci al confronto appartenenti a questa categoria comparano le impronte digitali in termini delle caratteristiche estratte dal pattern formato dalle creste;

## 10.2 Fingerprint : estrazione dell'impronta

1. **Stima dell'orientamento:** questo passaggio mira a determinare l'orientamento delle creste e delle scanalature presenti nell'impronta digitale. L'orientamento può variare in diverse parti dell'impronta digitale, quindi la stima è necessaria per ottenere una rappresentazione coerente. Da in output l'**orientation field**(campo d'orientamento);
2. **Localizzazione dell'impronta digitale:** questo passaggio consiste nel individuare e isolare l'area dell'impronta digitale all'interno di un'immagine più grande. Poiché l'impronta digitale può essere circondata da rumore o informazioni irrilevanti, la localizzazione dell'impronta digitale aiuta a ridurre l'area di interesse e concentra l'analisi sulle caratteristiche rilevanti. Da in output la nostra **regione d'interesse**;

3. **Estrazione delle creste:** una volta localizzata l'impronta digitale, questo passaggio mira a estrarre le creste delle scanalature. Le creste corrispondono alle linee continue presenti nell'impronta digitale. Questa estrazione può avvenire utilizzando algoritmi di elaborazione dell'immagine che rilevano i cambiamenti di intensità lungo le scanalature. Da in output l'immagine con solo l'estrazione delle creste;
4. **Assottigliamento:** dopo l'estrazione delle creste, è comune applicare un processo di assottigliamento. Questo passaggio riduce la larghezza delle creste per ottenere una rappresentazione più coerente e precisa dell'impronta digitale. Gli algoritmi di assottigliamento rimuovono i pixel aggiuntivi lungo le creste, mantenendo solo le linee centrali delle scanalature. Da in output le creste assottigliate;
5. **Estrazione delle minuzie:** le minuzie sono punti specifici di interesse lungo le creste dell'impronta digitale. Questi punti possono includere biforcati (dove una cresta si divide in due) e terminazioni (dove una cresta termina). La loro estrazione avviene utilizzando algoritmi che rilevano e identificano questi punti di interesse;
6. Si procede poi a creare un template partendo dalle minuzie, normalmente si cerca di creare un vettore delle caratteristiche che può contenere informazioni come la loro posizione o la loro direzione.

### 10.3 Fingerprint : verifica dell'impronta

1. **Sensing (acquisizione) delle impronte digitali:** la verifica delle impronte digitali inizia con l'acquisizione dell'immagine dell'impronta digitale utilizzando un sensore o uno scanner dedicato. Questo processo cattura l'immagine delle creste e delle scanalature presenti nell'impronta digitale;
2. **Allineamento dell'immagine delle impronte digitali:** dopo l'acquisizione, l'immagine delle impronte digitali potrebbe richiedere un allineamento per garantire che sia correttamente posizionata rispetto al template memorizzato. Questo passaggio può coinvolgere la correzione di rotazioni, traslazioni o altre distorsioni presenti nell'immagine;
3. **Estrazione delle minuzie:** successivamente, le minuzie vengono estratte dall'immagine delle impronte digitali. Questo processo coinvolge l'individuazione e l'estrazione dei punti di interesse unici, come le biforcati e le terminazioni, che sono ritenute rilevanti per la verifica delle impronte digitali;
4. **Filtraggio di Gabor:** Dopo l'estrazione delle minuzie, le immagini delle impronte digitali possono essere sottoposte a filtraggio di Gabor. Il filtraggio di Gabor è una tecnica che consente di evidenziare le caratteristiche salienti, come le linee e le texture, presenti nelle impronte digitali. Questo passaggio aiuta a migliorare la qualità e la distinzione delle caratteristiche dell'impronta digitale;
5. **Matching impronte digitali:** Il confronto delle impronte digitali avviene confrontando le minuzie estratte dall'immagine dell'impronta digitale con il template memorizzato.

### 10.4 Vantaggi e svantaggi

I **vantaggi** del fingerprint è la sua universalità, l'unicità, la permanenza, la facile misurabilità, l'accettabilità , efficacia, e l'acquisizione che può essere effettuata sia attivamente che passivamente ed ha un basso costo.

Gli **svantaggi** sono la sua poco efficienza infatti abbiamo bisogno tra i 256 bytes 1.2k per memorizzare l'impronta con una risoluzione di 500 dpi, anche i tempi di ricerca sono lunghi. In più è facilmente modificabile grazie ad abrasioni, rughe, sporcizia, sudore o disidratazione. In più è facilmente spoofabile.

### 10.5 Physical spoofing: Metodo di matsumoto

Il metodo di Matsumoto è una tecnica di falsificazione delle impronte digitali. Sfrutta le vulnerabilità dei sensori di impronte digitali ottici per creare falsi positivi durante la verifica delle impronte digitali.

L'attaccante crea una copia falsa di un'impronta digitale utilizzando materiali come la gelatina di silicone o la gomma. L'attaccante applica questa copia falsa sul suo dito e la presenta al sensore di impronte digitali ottico per la verifica. Il sensore di impronte digitali ottico utilizza la luce per acquisire l'immagine dell'impronta digitale. Tuttavia, questo tipo di sensore può essere ingannato dall'impronta digitale falsa creata dall'attaccante, poiché i materiali utilizzati nella copia falsa possono trasmettere la luce in modo simile al tessuto umano. Inoltre, l'attaccante può sfruttare la natura bidimensionale dei sensori di impronte digitali ottici, poiché spesso non sono in grado di rilevare la profondità delle scanalature dell'impronta digitale. Di conseguenza, anche una copia piatta dell'impronta digitale può essere sufficiente per ingannare il sensore.

Con i nuovi sensori questo tipo di attacco è facilmente bloccabile perché viene effettuato il **test di vitalità**:

1. Se la temperatura corporea non è superiore ai 36° l'autenticazione è rifiutata;
2. se non viene percepita la pressione arteriosa l'autenticazione è rifiutata;
3. se non viene percepito il numero di battiti cardiaci l'autenticazione è rifiutata.

# Multibiometric Systems

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.

## Vari sistemi multi-biometrici

- **Multiple algorithms:** utilizzo di più algoritmi o metodi di elaborazione delle biometrie per ottenere una maggiore accuratezza o robustezza nelle prestazioni;
- **Multiple sensors:** utilizzo di più sensori biometrici, come sensori di impronte digitali, sensori di riconoscimento del viso o sensori di riconoscimento dell'iride, per acquisire diverse modalità biometriche;
- **Multiple traits multimodal:** Utilizzo di più tratti biometrici, come impronte digitali, volti, iridi o vocali, per creare un sistema multimodale che combini più modalità biometriche per un'identificazione o autenticazione più accurata;
- **Multiple instances:** utilizzo di più istanze di un tratto biometrico specifico, come diverse immagini di impronte digitali dello stesso dito, per migliorare l'accuratezza e la robustezza delle biometrie;
- **Repeated instances:** utilizzo di più acquisizioni dello stesso tratto biometrico in momenti diversi per ottenere una rappresentazione più completa e stabile delle caratteristiche biometriche.

## 11.1 Multibiometria

- **Unica biometria e Multiplo sensore:** sono combinate le informazioni dello stesso tratto biometrico acquisito però con due tipi di sensori diversi;
- **Multipla biometria:** sono combinate le caratteristiche di due biometrie diverse, un volto ed un'impronta;
- **Unica biometria e Multipla unità:** si combinano due elementi diversi appartenenti alla stessa classe biometrica (l'impronta di due dita diverse);
- **Unica biometria e Multipla acquisizione:** si combinano più sessioni di acquisizione dello stesso tratto biometrico (più impressioni della stessa impronta).

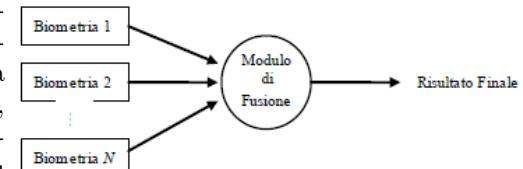
La scelta del numero e del tipo di biometria da valutare dipende soprattutto dalla natura del sistema. Un **sistema governativo** ha esigenze di sicurezza e di autenticità dell'utente superiori rispetto ad un sistema commerciale; la scelta di una biometria più affidabile quale l'iride o l'impronta digitale è favorita. Un **sistema embedded** su telefonini o computer palmari privilegia l'uso di biometrie rilevabili con l'hardware in dotazione al dispositivo, quindi voce, volto o firma.

## 11.2 Architettura del sistema

Un sistema multimodale può essere progettato secondo tre distinte architetture:

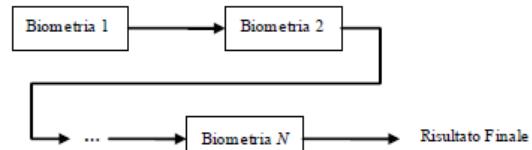
### Progettazione in parallelo

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.



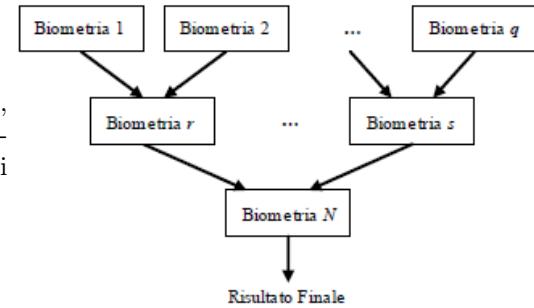
## Progettazione in serie

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.



## 11.3 Livelli e strategie di fusione

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

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

Di seguito saranno elencati i vari livelli di fusione:

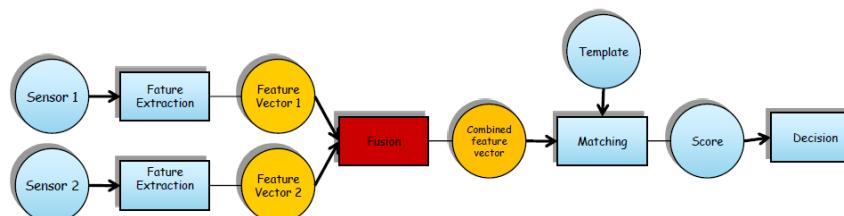
### 11.3.1 Fusione a livello di sensori

La fusione sui sensori è una tecnica utilizzata per combinare le informazioni provenienti da diversi sensori al fine di migliorare l'accuratezza, la robustezza e le prestazioni complessive del sistema.

### 11.3.2 Fusione a livello di feature

Le feature che sono state estratte con tecniche possibilmente diverse possono essere fuse per creare un nuovo vettore di caratteristiche per rappresentare l'individuo.

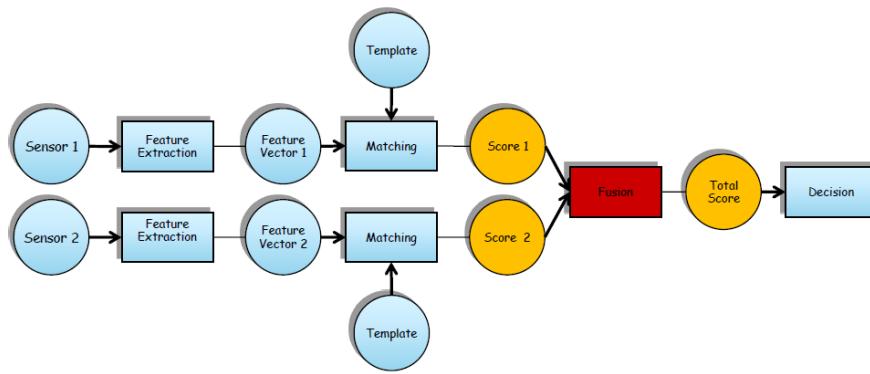
Si prevede un miglioramento dei risultati poiché sono presenti molte più informazioni. Possono generarsi dei problemi oppure si possono unire set di feature incompatibili. Infatti la combinazione di vettori di caratteristiche può causare "difformità della dimensionalità". Potrebbe quindi essere necessario un match più complesso. La combinazione di vettori può portare a includere dati rumorosi e/o ridondanti.



### 11.3.3 Fusione a livello di score

Diversi algoritmi di matching restituiscono un insieme di punteggi che sono fusi per generare un singolo punteggio finale. Abbiamo due tipologie di sistemi:

- **Transformation-based:** i punteggi di diversi match vengono prima normalizzati (trasformati) in un dominio comune e poi combinati usando le regole di fusione;
- **Classifier-based:** i punteggi di diversi classificatori sono considerati funzioni e sono inclusi in un vettore di caratteristiche. Un classificatore binario viene addestrato per discriminare tra vettori di punteggi autentici e impostori (NN-Neural Networks, SVM - Support Vector Machine).

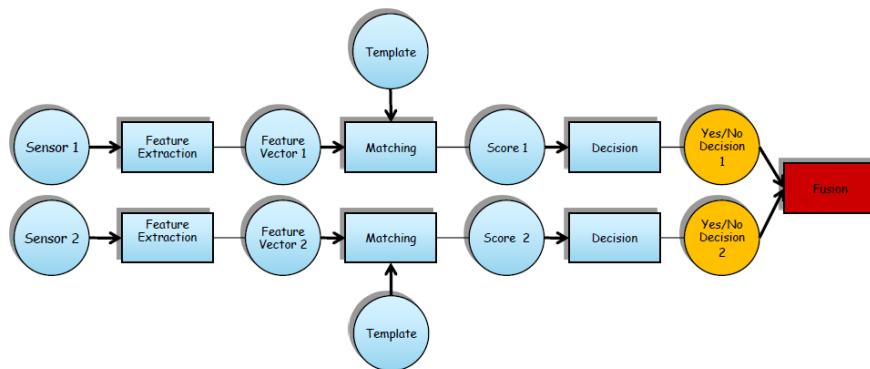


### 11.3.4 Fusione a livello di decisione

Ogni classificatore emette la propria decisione (accetta/rifiuta per verifica o per identificazione). La decisione finale viene presa combinando le singole decisioni secondo una regola di fusione.

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

- **Combinazione seriale di AND:** richiede l'autenticazione globale, tutte le decisioni devono essere positive. Migliora il FAR;
- **Combinazione parallela di OR:** l'utente può essere autenticato anche per una singola modalità biometrica. Migliora il FRR;
- Un'ulteriore importante regola di fusione a livello decisionale è **Majority Voting:** nel Majority Voting, ogni modalità biometrica o classificatore fornisce una decisione indipendente, ad esempio "autentico" o "non autentico". La decisione finale viene quindi determinata selezionando la classe che riceve la maggioranza dei voti.



## 11.4 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.

## 11.5 Strategia 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) quindi 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.



**IMPORTANTE!**

**DISCLAIMER**

Il materiale contenuto nel drive è stato raccolto e richiesto tramite autorizzazione ai ragazzi frequentanti il corso di studi di Informatica dell'Università degli Studi di Salerno.

Gli appunti e gli esercizi nascono da un uso e consumo degli autori che li hanno creati e risistemati, per tanto non ci assumiamo la responsabilità di eventuali mancanze o difetti all'interno del materiale pubblicato.

Il materiale sarà modificato aggiungendo il logo dell'associazione, in tal caso questo possa recare problemi ad alcuni autori di materiale pubblicato, tale persona può contattarci in privato ed eliminaremo o modificheremo il materiale in base alle sue preferenze. Ringraziamo eventuali segnalazioni di errori così da poter modificare e fornire il miglior materiale possibile a supporto degli studenti.

Associazione CoScienze