INTRODUZIONE

Lezione 1

I sistemi biometrici possono operare in due diverse modalità: *identification* e *verification*:

- ***Identification***: nessuna pretesa da parte dell'utente. Il sistema deve determinare la corrispondenza con uno dei soggetti della galleria di sistema mediante un'operazione di abbinamento *1:N*. Ritorna un possibile risultato = *identità riconosciuta*:
  - ***Open-Set***: il sistema determina se la *probe* p_i appartiene a un soggetto della galleria *G*. Alcune sonde potrebbero non appartenere a nessun soggetto in G -> il sistema ha un'opzione di rifiuto. Potrebbe rifiutare una probe appartenente a un soggetto iscritto.
  - ***Closed-Set***: tutte le *probe* appartengono a soggetti iscritti. Potrebbe ritornare un'identità errata.
  - ***Watch-list***: il sistema dispone di una lista di soggetti e verifica se la *probe* appartiene alla lista.
    - ***White-list***: ai soggetti in elenco è consentito l'accesso.
    - ***Black-list***: i soggetti in lista vengono rifiutati (possibile allarme).
- ***Verification***: una persona dichiara la sua identità. L'utente dice "*sono Andrea"* ed il sistema va a controllare se l'identità dichiarata, ovvero Andrea, è la stessa di quella presente in archivio. Quindi il suo riconoscimento diventa un processo di verifica (1:1) che richiede un match tra l'immagine rilevata (o i dati acquisiti) in tempo reale dai sensori e quella presente in un archivio.

INDICI DI PRESTAZIONE

Lezione 2

Diversi tipi di problemi possono sorgere all'interno dei sistemi biometrici:

- ***Wide intra-class variations***: il tratto biometrico potrebbe variare troppo all'interno di uno stesso soggetto;
- ***Very small inter-class variation***: il tratto biometrico potrebbe essere troppo simile tra due differenti soggetti;
- **Noisy and/or distorted acquisitions**;
- **Non universality**: il 4% della popolazione presenta impronte digitali di scarsa qualità. In alcuni gruppi è una caratteristica particolarmente diffusa (es. anziani);
- **Possible attacks (spoofing) in different moments**: ad esempio "copiare" le impronte digitali di una persona ed utilizzarle al posto suo.

ERRORI NELLA VERIFICA

In attività diverse possono verificarsi errori diversi. In verifica, un soggetto è *accettato* se la somiglianza (**Similarity Score**) ottenuta dall'abbinamento con i modelli di galleria corrispondenti all'identità dichiarata è ≥ della ***soglia di threshold***; altrimenti viene *rifiutato*.
Quindi possono verificarsi 4 casi:

- *L'identità dichiarata è vera ed il soggetto è accettato* (**Genuine Acceptance - GA** o **Genuine Match**);
- *L'identità dichiarata è vera ma il soggetto viene rifiutato* (**False Rejection - FR** o **False Non Match** o **type I error**);
- *Un soggetto impostore viene rifiutato* (**Genuine Reject - GR** o **Genuine Non Match**);
- *Viene accettato un soggetto impostore* (**False Acceptance - FA** o **False Match** o **type II error**).

Le misure più comuni per la verifica sono 5 e dipendono tutte dalla soglia di accettazione adottata (*threshold*) ***t***:

- **FAR** (**False Acceptance Rate**): probabilità di riconoscere erroneamente una *persona <u>non</u> registrata come persona registrata*, $FAR = \frac{False\ Acceptance}{(False\ Acceptance + Total\ Rejection)}$;
- **FRR** (**False Rejection Rate**): probabilità di riconoscere erroneamente una *<u>persona registrata</u>* come *persona <u>non</u> registrata*, $FRR = \frac{False\ Rejection}{(False\ Rejection + Total\ Acceptance)}$;
- **ERR** (**Equal Error Rate**): è un punto in cui FAR e FRR si intersecano. Un dispositivo con EER inferiore è considerato più preciso.
- **DET** (**Detection Error Trade-off**): è un grafico dei tassi di errore per i sistemi di classificazione binaria che mette in corrispondenza il *False Acceptance Rate* con *False Rejection Rate.*
- **ROC** (**Receiving Operating Curve**): curva tracciata su un piano cartesiano dove gli assi sono il Genuine Acceptance Rate e il False Acceptance Rate. Rappresenta quindi la probabilità che un soggetto venga correttamente identificato al variare del FAR.

Tutti i campioni utilizzati per gli esperimenti di valutazione sono etichettati con l'identità corretta (*Ground Truth* - verità fondamentale). Definiamo:

- **id(template)**: una funzione che, dato un *template* (i dati archiviati), restituisce la sua vera identità;
- **topMatch(p_j , identity)**: una funzione che restituisce la migliore corrispondenza tra p_j ed i template associati all'identità dichiarata nella galleria;
- **s(t_1,t_2)**: una funzione che restituisce la somiglianza tra template t_1 e template t_2;
- **P_G**: insieme di *probe* appartenenti a soggetti in galleria;
- **P_N**: insieme di *probe* appartenenti a soggetti <u>non</u> presenti in galleria.

$$FAR = \frac{|\{p_j : s_{xj} \geq t \wedge id(g_x) \neq id(p_j)\}|}{|\{p_j : id(g_x) \neq id(p_j)\}|}$$

dove:

- p_j = template
- g_x = $topMatch(p_j, i)$ → l'utente che invia il modello p_j dichiara una falsa identità i che è diversa da quella restituita dalla funzione di verità fondamentale per p che è $id(p_j)$
- s_{xj} = $s(g_x, p_j)$ → ?????????
- Scenario 1: $p_j \in P_G \cup P_N \wedge i \in I$ → l'utente è registrato ma dichiara una falsa identità.
- Scenario 2: $p_j \in P_G \wedge i \in I$ → scenario 2: l'utente non è registrato.

NUMERATORE: numero di volte in cui la *claimed identity* viene accettata nonostante l'identità sia falsa.
DENOMINATORE: numero totale di volte in cui il probe aveva l'identità dichiarata diversa da quella reale. | è la cardinalità dell'insieme di persone che <u>non</u> dobbiamo considerare correttamente.

$$FRR = \frac{|\{p_j : s_{xj} \leq t \wedge id(g_x) = id(p_j)\}|}{|\{p_j : id(g_x) = id(p_j)\}|}$$

dove:

- g_x = $topMatch(p_j, id(p_j)$ → $id(p_j)$ viene utilizzato al posto di una i generica per sottolineare che l'affermazione è *genuina*.

- $s_{xj} = s(g_x, p_j)$ → ????????
- $p_j \in P_G$ → utente *registrato* in galleria.

NUMERATORE: numero di volte in cui avviene la *false reject*.
DENOMINATORE: numero totale di volte in cui il probe aveva l'identità dichiarata uguale a quella reale. | è la cardinalità dell'insieme di persone che dobbiamo considerare correttamente.

- **Genuine Acceptance Rate**: 1 - F AR.
- **Equal Error Rate**: punto d'intersezione quando *FAR = FRR.*
- **Zero FRR**: punto su *FAR* quando *FRR* vale 0.
- **Zero FAR**: punto su *FRR* quando *FAR* vale 0.
- **Detection Error Trade-off**: raffigura *FRR vs FAR*, tracciato in forma logaritmica.
- **Reception operating curve**: raffigura *GAR vs FAR*.

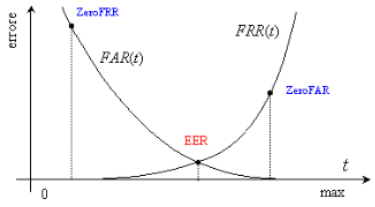


Figure 1: ERR, Zero FRR and Zero FAR.

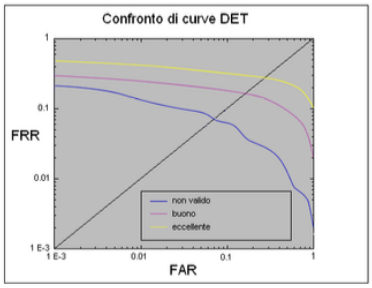


Figure 2: DET curves.

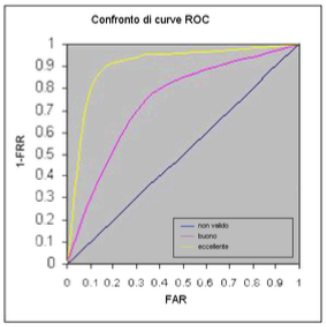
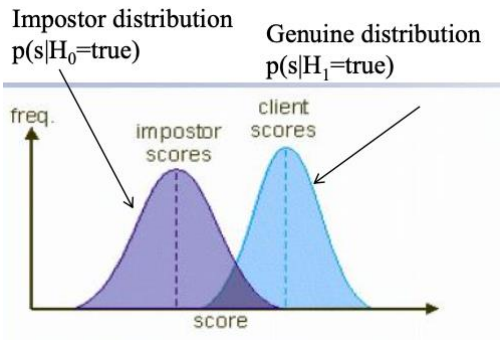
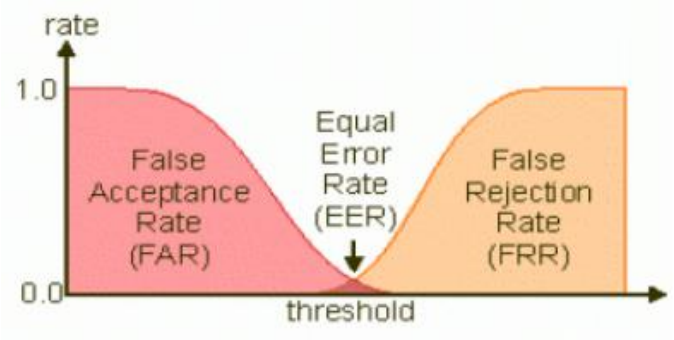


Figure 3: ROC curves.

Un punteggio è detto ***genuine*** (***autentico***) se risulta dall'abbinamento di due campioni del tratto biometrico di uno stesso individuo registrato; si dice ***impostor*** se risulta dall'abbinamento del campione di un soggetto non registrato.
Allora, date le ipotesi **H0** (*persone diverse*) e **H1** (*stessa persona*) e le possibili decisioni **D0** (*persone diverse*) e **D1** (*stessa persona*), si ha **FAR = p(D1|H0 = vero)** (probabilità che siano la stessa persona t.c. sono persone diverse) e **FRR = p(D0 |H1 = vero)** (probabilità che non siano la stessa persona t.c. sono la stessa persona). Guarda la figura





ERR = punto in cui FAR e FRR si intersecano

Vale la pena sottolineare che la soglia di accettazione è cruciale e dipende dalle esigenze applicative. Considerando le distanze:

- *Una soglia troppo bassa causa molti errori di tipo I **(FRR alto)***;
- *Una soglia troppo bassa causa molti errori di tipo II (**FAR alto**)*.

Quindi, una scelta popolare è quella di impostare la soglia corrispondente a ERR.

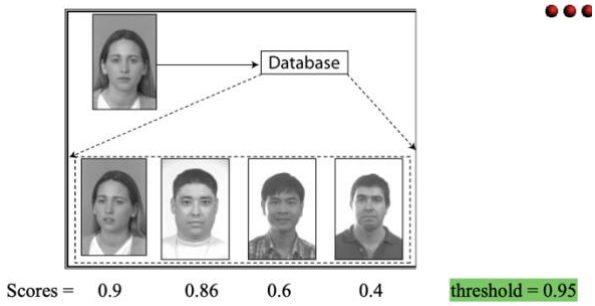
ERRORI NELL'IDENTIFICAZIONE – OPEN SET

Non è noto se il soggetto presentato al sistema biometrico per il riconoscimento sia iscritto al sistema o meno, quindi se è presente o meno in galleria. Pertanto, il sistema deve decidere se rifiutarlo o riconoscerlo come uno dei soggetti iscritti. È l'opposto di "*Closed-Set identification*".
L'individuo non fa una rivendicazione di identità. L'obiettivo del sistema è determinare se il *soggetto probe* è presente nel database e, in caso affermativo, chi è il *soggetto probe*.
Ci sono quindi più possibili situazioni di errore, a seconda del valore di threshold di riconoscimento (punteggio/somiglianza/distanza).

Possibili scenari:

1. Ci sono alcuni individui sopra la soglia (***rilevamento corretto → allarme***) e l'individuo con il punteggio più alto è quello giusto (***identificazione corretta***);
2. Non c'è nessun individuo al di sopra della soglia (***nessun rilevamento → nessun allarme***), quindi non ci interessa guardare l'individuo in alto (***nessuna identificazione corretta***);
3. Ci sono alcuni individui sopra la soglia (***rilevamento corretto → allarme***) ma l'individuo con il punteggio più alto non è quello giusto (***nessuna identificazione corretta***);

Esempio:



Questo è il punto 2.

Definiamo $rango(p_j)$ come posizione nella lista dei *similarity score* dove è presente il primo template dell'identità corretta.

Misure di errore comuni:

- Il ***Detection and Identification Rate*** (***DIR***) ***at rango k*** è la probabilità con cui gli individui, che si trovano in un database, sono correttamente identificati in un'identificazione open-set, al rango k (il soggetto corretto viene riportato in posizione k), data dal rapporto tra il numero di *individui correttamente riconosciuti al rango k* e *il numero di sonde appartenenti agli individui in* P_G:

$$DIR(t,k) = \frac{|\{p_j : rango(p_j) \leq k \wedge s_{i,j} \geq t \wedge id(g_i) = id(p_j)\}|}{|P_G|} \quad \forall p_j \in P_G$$

  Il miglior risultato che si può ottenere è $DIR(t,k) = 1$.

- Il ***False Reject Rate*** (***FRR***) è la probabilità di false reject, al rango 1, espressa come:

$$FRR(t) = 1 - DIR(t,1)$$

  È il complemento di DIR

- Il ***False Acceptance Rate*** o ***False Alarm Rate*** (***whatch list***) è dato dal numero di volte in cui il sistema restituisce un allarme errato, eseguendo prove con probe appartenenti a soggetti

non presenti nel database (***in P_N***), come nel *terzo scenario*. Cioè è il rapporto tra il numero di impostori riconosciuti per errore e il numero totale di impostori in P_N:

$$FAR(t) = \frac{|\{p_j : \max_i s_{ij} \geq t\}|}{|P_N|} \quad \forall p_j \in P_N \; \forall g_i \in G$$

- L'***Equal Error Rate*** è il punto in cui i due errori di probabilità sono uguali, ad es. :

$$EER = \{x : FRR(t) = x \wedge FAR(t) = x\}$$

- La ***Open-set*** (***Watchlist***) ***Receiver Operating Characteristic*** (***ROC***) rappresenta DIR rispetto a FAR ed è utile per trovare il *threshold migliore*, in modo tale che DIR sia massimizzato e FAR ridotto al minimo. È necessario un compromesso poiché le due misure dipendono entrambe dal threshold del punteggio, ma in direzioni opposte:
  - Se alziamo la soglia, diminuisce il FAR, ma diminuisce anche il DIR;
  - Se abbassiamo la soglia, aumenta il DIR, ma aumenta anche il FAR.

Con *watchlist* indico l'Open-Set Identification, lo uso come sinonimo.

La selezione di una *watchlist threshold* dipenderà dal *tipo di applicazione*:

- ***Applicazioni che richiedono falsi allarmi estremamente bassi***: quando un allarme richiede un'azione immediata, ciò potrebbe causare disturbo e confusione. Inoltre, un allarme e la successiva azione possono rendere evidente che la sorveglianza viene eseguita, e possono ridurre al minimo la possibilità di catturare un futuro sospetto.
- ***Applicazioni che richiedono un'altissima probabilità di rilevamento e identificazione***: la preoccupazione principale è individuare qualcuno nella *watchlist* (lista di controllo); i falsi allarmi sono una preoccupazione secondaria e verranno affrontati secondo procedure predefinite.
- ***Applicazioni che richiedono un basso livello di falsi allarmi e rilevamento/identificazione***: la preoccupazione principale è la riduzione dei falsi allarmi ed è accettabile gestire un livello basso di rilevamento/identificazione.
- ***Applicazioni che richiedono un alto livello di falsi allarmi e rilevamento/identificazione***: la preoccupazione principale sono le prestazioni di rilevamento/identificazione più elevate ed è accettabile gestire anche un FAR elevato.
- ***Applicazioni che non richiedono soglie***: l'utente desidera per l'indagine tutti i risultati con le corrispondenti misure di confidenza.

ERRORI NELL'IDENTIFICAZIONE – CLOSED SET

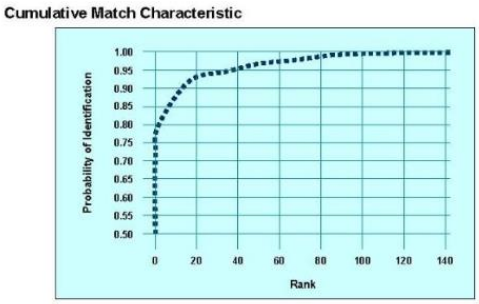
Qualsiasi soggetto presentato al sistema biometrico per il riconoscimento è noto per essere iscritto nel sistema, quindi, presente nella galleria; pertanto, in linea di principio non è necessario alcun rifiuto a meno che la qualità del tratto biometrico in ingresso non sia troppo bassa per essere elaborata. È l'opposto di "*Open-Set Identification*".

Possibili misure:

- Il ***Cumulative Match Score*** (*CMS*) ***al rango k*** è la probabilità di identificazione al rango k: eseguendo molte prove con soggetti diversi, sapremo con quale frequenza il sistema restituirà un risultato corretto con il punteggio di similarità nelle prime *k* posizioni. È espresso come rapporto tra il numero di individui correttamente riconosciuti tra i primi *k* e

il numero totale di individui nel test set (probe). CMS al rank 1 è anche detto RR (Recognition rate)

- La curva ***Comulative Match Characteristic*** (***CMC***) curva tracciata su un piano cartesiano dove gli assi sono la *Probability of Identification* e il *Rank k*. Rappresenta quindi la probabilità che un soggetto venga correttamente identificato al variare del Rank k.



INDICI DI PRESTAZIONE - CONTINUA

Lezione 2Bis

Prima di tutto, ricordiamo che durante l'analisi statistica offline conosciamo la ground truth, ovvero tutti i campioni sono etichettati (*labelled*). Questo non è vero nelle operazioni in tempo reale.
Durante la valutazione delle prestazioni, ogni campione presentato come probe può svolgere il ruolo di un *campione autentico* o di un *impostore*, a seconda della galleria per l'esperimento in questione e dell'eventuale rivendicazione di identità attribuita alla probe se in modalità di *verification*.

Per organizzare il dataset per gli esperimenti devono essere prese delle decisioni:

- Per prima cosa scegliere in che modo dividere il dataset in ***training set*** (***TR***) e ***test set*** (***TS***):
  - Per garantire la generalizzazione, nel set di formazione devono essere inclusi modelli di qualità diversa;
  - Il TS contiene alcune identità che non compaiono nel TR, per testare meglio la generalizzabilità;
  - Non è consentita la sovrapposizione tra TR e TS;
- La seconda scelta è quali modelli inserire nel set di *probe P* e quali nel set di *galleria G*:
  - I modelli di migliore qualità vengono solitamente inseriti nella galleria poiché l'iscrizione viene generalmente effettuata in condizioni controllate (quindi immagini femre e ben visibili, non fatte in movimento), ma sono possibili altre scelte;
  - Non è consentita alcuna sovrapposizione tra P e G (i due insiemi possono contenere modelli diversi delle stesse identità);
- La terza scelta è se tutti i template nel set di probe devono appartenere ai soggetti della galleria ($\boldsymbol{P = P_G}$) o no ($\boldsymbol{P = P_G \cup P_N}$):
  - In verifica questa scelta non incide sui risultati;
  - In *identification closed set* questa scelta non è possibile;
  - In *identification open set* questa scelta può influenzare i risultati in base al rapporto tra il numero di soggetti in galleria e il numero di probe totali;
  - Questa scelta è basata sui soggetti.

Per quanto riguarda la ***validation*** dovremmo partizionare il set di dati in modi diversi, ripetere la validation e prendere prestazioni medie, al fine di evitare il *bias* dovuto a una scelta specifica degli elementi di partizione. Nella ***k-fold cross-validation*** i dati sono divisi in ***k*** sottoinsiemi, quindi il modello viene applicato ***k*** volte utilizzando un sottoinsieme come insieme di *convalida* e gli altri come *train set*. La stima dell'errore viene mediata su tutte le *k* prove. Come regola generale, è generalmente preferito ***k=5*** o ***10***.

ALL VS ALL – PROBE VS GALLERY E DISTANCE MATRIX

Per ogni coppia *probe-gallery*, è possibile calcolare preventivamente una matrice di distanza ALL_{PROBE} VS $ALL_{GALLERY}$, memorizzando tutte le distanze tra le coppie di modelli (un modello di probe rispetto a un modello di galleria).
Ogni riga della matrice corrisponde ad un'operazione di riconoscimento su un probe in ingresso ed ogni partizione probe-gallery produce una matrice di distanza diversa.

La matrice delle distanze può essere utilizzata per la valutazione delle prestazioni di tutti i tipi di applicazioni come la *verification*, l'*Open-Set Identification* e la *Closed-Set Identification*:

- ***Verification***:
  - Solo i modelli in *gallery* appartenenti all'identità dichiarata vengono confrontati con il *probe*.
  - Non è importante chi c'è nella *gallery*, ma l'identità rivendicata.
  - Ogni riga è etichettata con l'identità della *ground truth probe* (ad esempio *ID A*) e, per la verifica, con l'identità dichiarata (ad esempio *Claim A*).
  - La valutazione delle prestazioni può essere ottenuta separando chiaramente *probe* e *gallery* (diversi sottoinsiemi di modelli) e veri e propri impostori (i modelli che svolgono il ruolo di impostori sono associati alla rivendicazione di un'identità diversa).

ESEMPIO:

Example: identities A, B, C, D, E, F → in the **dataset**
identities A, B, C, D → in the **gallery** with a single istance
identities E, F → play the role of **impostors in all cases**
d() = distance form the probe t = acceptance threshold

	Probes	A1	B1	C1	D1
ID A – Claim A	P1	1	4	2	3
ID D – Claim C	P2	4	1	3	2
ID E – Claim D	P3	4	2	1	3
ID C – Claim C	P4	…	…	…	…
ID F – Claim B	P5	…	…	…	…

Example of distance matrix (only **order** of values is shown)

Each row is **a single** verification operation

La *probe* ***P1*** con identità ***A*** claim ***A*** (*reclama l'identità* ***A***). Questa è una *GENUINE CLAIM* (e lo sappiamo granzie al *ground truth*) e deve essere accettata dal sistema. Se ho una distanza ***d(A1) <= t*** (d(A1) = 1 è la distanza) allora avrò una ***GenuineAcceptance***, se invece ho **d(A1) > t** allora in QUESTO caso avrò una ***FalseRejection*** (poichè l'identità è corretta ma ha sbagliato il sistema). In questo caso per ottenere una GA devo settare un valore di *threshold* ***t = 1***.
P2 appartiene all'identità ***D*** ma l'identity claim è ***C***. ***D*** è in galleria ma reclama un'altra identità. Ovviamente se ***d(C1) > t*** abbiamo una ***GenuineRejection*** ed il sistema ha captato efficacemente l'impostore, altrimenti abbiamo una ***FalseAcceptance***.

La distance con la claimed identity (***C1***) è 3. Se setto ***t = 2*** allora ho una ***GR***, in questo modo ***C1 > t***.

- ***P1* caso GENUINE:**
  - *$d(A_1)$ <= t → GA++*
  - *$d(A_1)$ > t → FR ++*
- ***P2* IMPOSTORE (il soggetto <u>è in galleria ma rivendica un'altra identità</u>):**
  - *$d(C_1)$ <= t → FA++*
  - *$d(C_1)$ > t → GR++*
- ***P3* IMPOSTORE (soggetto <u>non in galleria</u>)**:
  - *$d(D_1)$ <= t → FA++*
  - *$d(D_1)$ > t → GR++*

Incrementando il numero di *samples* per soggetto si va a decrementare il FRR (perché abbiamo più possibilità di riconoscere le genuine identity) ma aumenta il FAR (poiché aumenta la possibilità che un False probe sia simile ad una Genuine Identity).
È necessario effettuare un numero sufficiente di valutazioni per calcolare un risultato medio attendibile: ogni volta probe set e gallery set sono scelti in modo diverso, e c'è una possibile diversa distribuzione di probe autentici e impostori;

- ***Identification – <u>Open Set</u>***:
  - In questo caso il probe da identificare *potrebbe <u>non</u> essere nel sistema*.
  - Ogni riga è etichettata solo con l'identità ground truth (ad es. *A*) della sonda (nessuna rivendicazione, infatti <u>non c'è</u> *Claim A*.).
  - Le *probe autentiche* sono quelle appartenenti alle identità nella galleria, mentre le *probe impostori* sono quelle appartenenti alle identità <u>non</u> nella galleria.

Esempio:

Example: identities A, B, C, D, E, F in the **dataset**
identities A, B, C, D in the **gallery** with a single istance
identities E, F play the role of **impostors**
d() = distance form the probe t = acceptance threshold

	Probes	A1	B1	C1	D1
<u>A</u>	P1	1	4	2	3
<u>D</u>	P2	4	1	3	2
<u>E - Impostor</u>	P3	4	2	1	3
<u>C</u>	P4	…	…	…	…
<u>F - Impostor</u>	P5	…	…	…	…

Example of distance matrix (only **order** of values is shown)

Each row is a **single** identification operation

Il valore di threshold *t* è la distanza dalla probe che possiamo tollerare per accettare eventuali probe.

Per ogni P_n viene ordinata la lista delle distanze, come ad esempio per P_1 = **<u>$d(A_1)$</u>**, $d(C_1)$, $d(D_1)$, $d(B_1)$. La *probe **P1*** con identità ***A*** claim ***A*** (*reclama l'identità **A***). Questa è una *GENUINE CLAIM* e deve essere accettata dal sistema. Se ***$d(A_1)$ <u><=</u> t*** allora incremento il ***DI(1,t)***, ovvero incremento il ***Detect and Identification outcomes at rank 1*** per il treshold ***t***, mentre se *$d(A_1)$ > t* avrò un *False Rejection*.

Anche in questo caso è importante effettuare un numero sufficiente di valutazioni, cambiando set di sondaggi e gallerie e distribuzione di probe autentiche e impostori.

- ***Identification – <u>Closed Set</u>***:
  - La probe da identificare *appartiene <u>sempre</u> ad un soggetto inserito nella galler*ia (infatti non c'è *E - impostor*);
  - Tutti i modelli della gallery vengono confrontati con il probe;
  - Ogni riga è etichettata con l'identità di verità del probe;
  - Nessun impostore compare negli esperimenti (<u>tutte</u> le identità sono nella gallery);
  - La valutazione delle prestazioni può essere ottenuta separando chiaramente probe e gallery;
  - **<u>Non c'è un treshold *t* di accettazione</u>**

Esempio con un solo 1 istanza per identities in gallery:

Example: identities A, B, C, D, E, F in the **dataset**
identities A, B, C, D, E, F in the **gallery** with a single istance
d() = distance form the probe

Tutte le identities sono nella gallery!

	Probes	A1	B1	C1	D1	E1	F1
A	P1	1	4	2	3	6	5
D	P2	6	1	4	2	3	5
E	P3	5	2	1	3	4	6
C	P4	…	…	…	…		
F	P5	…	…	…	…		

Example of distance matrix (only **order** of values is shown)

Each row is **a single** identification operation

Non c'è il valore di threshold *t* e tutte le probe sono in galleria.

Per ogni P_n viene ordinata la lista delle distanze, come ad esempio per P_1 = **<u>$d(A_1)$</u>**, $d(C_1)$, $d(D_1)$, $d(B_1)$, $d(F_1)$, $d(E_1)$.

Per P_1 il template corretto per A è A_1 e questo risultato contribuisce al ***CMS*** (Cumulative Match Score) al rank 1 ed è anche detto *Recognition Rate* (*RR*).

Per P_2 il template corretto per D è D_1 e questo risultato contribuisce al ***CMS*** (Cumulative Match Score) al rank 2.

Esempio con un solo 1 istanza per identities in gallery:

Example: identities A, B, C, D, E, F in the **dataset**
identities A, B, C, D, E, F in the **gallery** with multiple istances
d() = distance form the probe

	Probes	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F1	F2
A	P1	2	1	8	11	5	7	6	12	10	3	9	4
D	P2	11	3	2	7	8	10	5	1	6	12	9	4
E	P3	9	10	4	7	2	3	12	5	6	8	11	1
C	P4	…	…	…	…	…	…	…	…				
F	P5	…	…	…	…	…	…	…	…				

Example of distance matrix (only **order** of values is shown)

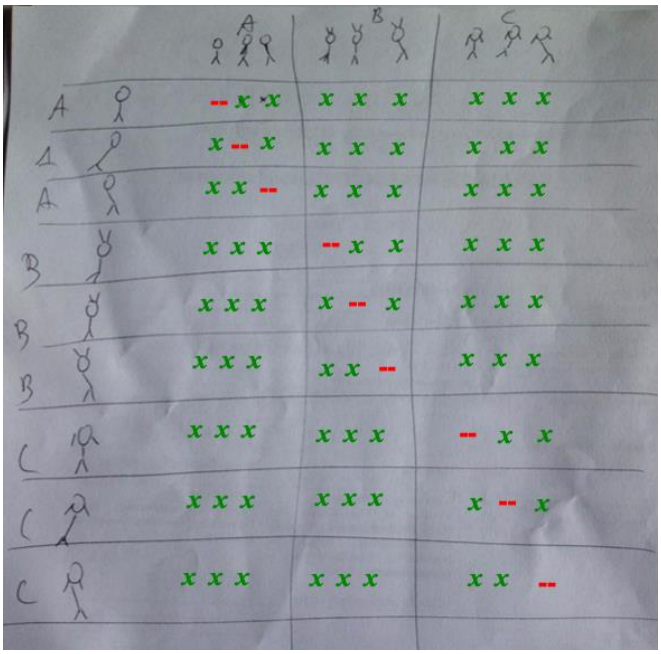
Each row is **a single** identification operation

Per ogni P_n viene ordinata la lista delle distanze, come ad esempio per P_1 = **<u>$d(A_2)$</u>**, $d(A_1)$, $d(E_2)$, $d(F_2)$, $d(C_1)$, $d(D_1)$, …

Per P_1 il template corretto per A è A_2 e questo risultato contribuisce al ***CMS*** (Cumulative Match Score) al rank 1 ed è anche detto *Recognition Rate* (*RR*).

Per P_2 il template corretto per D è D_2 e questo risultato contribuisce al ***CMS*** (Cumulative Match Score) al rank 1 ed è anche detto *Recognition Rate* (*RR*). In questo caso l'identità in prima posizione (nella lista delle distanze ordinate per P_2), D_2, è quella giusta ed il nuovo template per D ha migliorato il riconoscimento. Il vecchio template era D_1 nell'esempio precedente.

È possibile calcolare statistiche più comprensive utilizzando una matrice di distanza completa nel senso che non solo calcoliamo solo l'intera matrice ma consideriamo anche ogni riga come un diverso set di esperimenti. Differenti nel senso ogni riga rappresenta più di un aspetto.



"***x***" indica *valid matching operation* e "**-**" indica *invalid matching operation*. Ogni lettera A, B, C, indica un'*identity*.

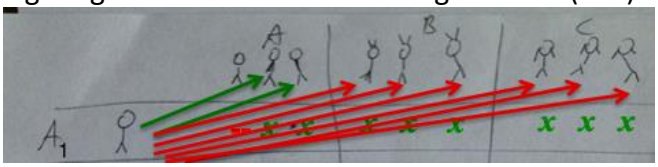
Una possibile strategia di utilizzo della matrice di distanza ALL vs. ALL è la seguente:
• *Ogni probe (righe) è considerata a sua volta genuina o impostore*;
• *I risultati vengono accumulati per ottenere le statistiche finali*;
• *Ogni riga contribuisce eventualmente più volte in base al numero di esperimenti che può eventualmente rappresentare (gli esempi sono i modelli passati al sistema per addestrarlo)*;
• *Per semplicità si può ipotizzare lo stesso numero di campioni per soggetto (nel nostro caso 3 per ogni set)*.

Alcune definizioni:

- **M** = distance matrix
- **N** = numero di subject (insieme di omini come ad esempio A con i tre personaggi)
- **S** = numero di templates per subject (il numero di omini per subject)
- **|G|** = numero totale di sample (|G| = SxN)
- **i** = indice di riga (*probe templates*) e ***label(i)*** è l'identità associata
- **j** = indice di colonna (*gallery template*) e ***label(j)*** è l'identità associata
- **label(·)** = ground truth

Verification con single-templates:

- Ogni riga è un insieme di **|G| - 1** operazioni
- Ogni riga contiene S – 1 tentativi genuine e (N-1) x S tentativi impostor.



In verde gli S-1 [3-1] tentativi genuine
In rosso gli (N-1) x S [(3-1) x 3 = 6] tentativi impostori.

- Numero totale di tentativi genuine TG = |G| x (S-1) [TG = |9| x (3-1) = 18]
- Numero totale di tentativi impostor TI = |G| x (N-1) x S [TI = |9| x (3-1) x 3 =]

```
for each threshold t:
    for each cell M_{i,j} with i ≠ j:
        if M_{i,j} ≤ t:
            if label(i) = label(j): GA++
            else: FA++
        else if label(i) = label(j): FR++
        else: GR++
    GAR(t) = GA/TG
    FAR(t) = FA/TI
    FRR(t) = FR/TG
    GRR(t) = GR/TI
```

RECOGNITION RELIABILITY

Lezione 3

Il compito di *identification* (soprattutto nell'Open-Set) è molto più difficile per i sistemi biometrici (ma anche per gli operatori umani) rispetto al compito di *verification*.

Misure come FAR, FRR, CMS, … non sono sufficienti per dare una valutazione approfondita degli algoritmi.
Per un confronto affidabile dobbiamo considerare:

- Numero e caratteristiche del database;
- Dimensioni dell'immagine;
- Dimensione del probe e della gallery;

Le Features estratte da un tratto biometrico sono caratteristiche dell'individuo e ne rappresentano il *Template*. Questi dati soffrono di problemi quali il *Template Aging* e dovrebbero essere sottoposti al *Template Updating*, che può essere di tipo *Supervised* (richiedere a un supervisore di assegnare etichette di identità ai dati appena acquisiti, spesso lavora offline) o *Semi-Supervised* (utilizzare l'unione di dati etichettati e non etichettati, lavora sia online che offline).

Il Doddington Zoo ha definito alcune similitudini ad animali nel contesto dei sistemi di riconoscimento:

- ***Pecora***: matchano bene con sé stessi e poco con quelli di altri (Genuine Acceptance);
- ***Capra***: matchano male con i loro stessi tratti (produce un alto tasso di False Rejection);
- ***Agnello***: possono essere facilmente impersonati (persone che facilmente si impersonano in altre persone in maniera involontaria - False Acceptance);
- ***Lupo***: sono bravi ad impersonare altri utenti (persone che facilmente si impersonano in altre persone in maniera volontaria - False Acceptance);

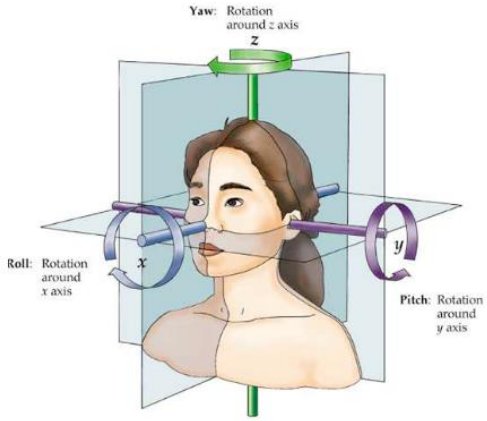
Succesivamente, Yager e Dunstone hanno definito altre similitudini in termini di punteggi sia *genuini* che *impostori* per quanto riguarda un utente, anziché solo uno di essi come i primi quattro sopra.

Si consideri una popolazione di utenti $\boldsymbol{P}$ e un insieme di *verification match score* $\boldsymbol{S}$.
Per ogni coppia di utenti $j, k \in P$, esiste un insieme $\{s\ (j,k)\} \subset S$ contenente tutti i risultati della verifica ottenuti abbinando uno dei i modelli di j contro un modello di riferimento appartenente a k. I punteggi *genuini* dell'utente k sono l'insieme $G_k = \{s\ (k,k)\}$ e i punteggi degli *impostori* di k sono l'insieme $I_k = \{s\ (j,k)\} \cup \{s\ (k,j)\}\ \forall j \neq k$.
Quindi, le nuove similitudini sono:

- ***Camaleonte***: appaiono sempre come altri utenti (Low False Rejection e High False Acceptance). Hanno un valore alto di G_k e I_k. Un esempio è un utente con caratteristiche generiche molto pesate dall'algoritmo di matching;
- ***Verme***: causano spesso errori di sistema, è il peggior utente. Ha un valore basso di G_k ed un valore alto di I_k. Si abbina male con se stesso e può avere un punteggio di corrispondenza elevato con gli altri,
- ***Fantasma***: soffrono di basse corrispondenze (È probabile che causi FR, ma non FA). Hanno un valore basso di G_k e I_k. Un esempio è un utente che ha problemi a registrarsi nel sistema;
- ***Colomba***: raramente incorrono in errori, è il miglior utente (Genuine Acceptance). Ha un valore alto di G_k ed un valore basso di I_k. È puro e riconoscibile, si abbina bene con se stesso e male contro gli altri. Un esempio sono persone con caratteristiche non comuni, come:
    - Persone con il colore degli occhi diversi (ad es. 1 marrone ed 1 azzurro);
    - Persone con un naso molto grande;
    - Persone con una forma particolare della testa;

I campioni di input possono avere qualità diverse e le procedure di riconoscimento possono avere una precisione diversa, quindi non tutte le risposte sono ugualmente affidabili.
Un possibile approccio si basa sulla *qualità dell'immagine*. Per misurare la qualità di un'immagine di training:

- Possiamo modellarlo creando un “average template” da tutte le facce, la cui qualità è considerata come un riferimento;
- Possiamo stimare la nitidezza dell'immagine in base alla mancanza di dettagli ad alta frequenza;
- Possiamo usare:
    - ***SP*** per misurare la distorsione rispetto alla posa frontale (disallineamento di roll, yaw and pitch) $\boldsymbol{SP = \alpha(1-roll) + \beta(1-yaw) + \gamma(1-pitch)}$.
    - ***SI*** per misurare l'omogeneità dei livelli di grigio in regioni facciali predeterminate $\boldsymbol{SI = 1 - F(std(mc))}$
    - ***SY*** per misurare la simmetria della faccia $\boldsymbol{SY = \sum_{(i,j)\in X} sym(Pi,Pj)}$.



Possiamo scartare i dati ininfluenti o superflui, ma un buon sistema biometrico è quello che garantisce i migliori risultati al minimo scarto, che ha dati ben distribuiti e che ovviamente ha buone FAR e FRR.
Un altro approccio si basa sui "*margini basati sulla stima dell'errore*"
Le *prestazioni del sistema* sono misurate in termini di:

$$FAR(\Delta) = \frac{\text{mumber of } FAs(\Delta)}{\text{number of impostor accesses}}$$
$$FRR(\Delta) = \frac{\text{mumber of } FRs(\Delta)}{\text{number of client accesses}}$$

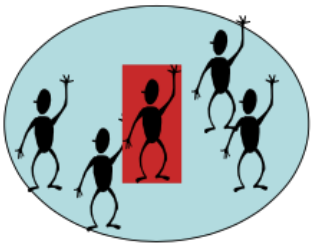
Il *margine* è definito come:

$$M(\Delta) = |FAR(\Delta) - FRR(\Delta)|$$

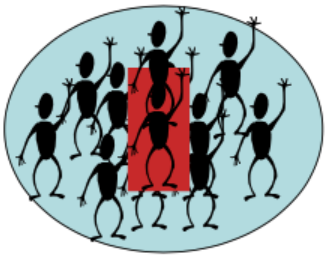
Un altro approccio ancora è "*System Response Reliability (SRR)*".
La *System Response Reliability* (*srr* $\in$ *[0,1]*) misura l'abilità del sistema nel dividere i *genuine* con gli *impostor* per ogni probe.
L'SRR si basa su diverse versioni della funzione φ. Abbiamo due diverse funzioni φ:

- *Distanza relativa*;
- *Density ratio*;

Entrambe le funzioni misurano la quantità di "*confusione*" tra i possibili candidati.
Misura quanto è "Crowded" (affollato) il soggetto, più è Crowded più è probabile che sia un impostore e si avrà una risposta meno attendibile di conseguenza.



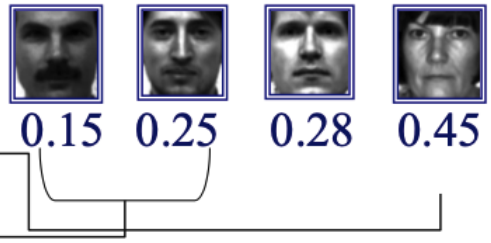
"less crowded" = Risposta più affidabile



"more crowded" = Risposta meno affidabile

Data una *probe* ***p*** ed un *sistema* ***A*** con *galleria* ***G***, ***la distanza relativa*** è definita come:

$$\varphi(p) = \frac{F(d(p,g_{i_2})) - F(d(p,g_{i_1}))}{F(d(p,g_{i_{|G|}}))}$$

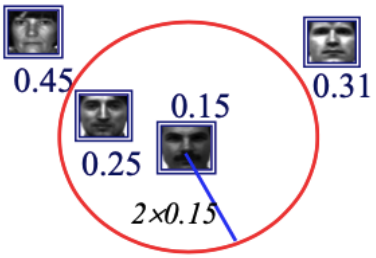


Data una *probe* ***p*** ed un *sistema* ***A*** con *galleria* ***G***, la ***density ratio*** è definita come:

$$\varphi(p) = 1 - |N_b| / |G|$$

With

$$N_b = \{g_{i_k} \in G \mid F(d(p,g_{i_k})) < 2 \cdot F((d(p,g_1)))\}$$



2×0.15 = 0.30

Density Ratio = 1 – 1/3
= 0.66 …

Occorre individuare un valore che favorisca una corretta separazione tra *errati respingimenti di soggetti iscritti* e *errati riconoscimenti di non iscritti*, entrambi supportati dal valore di attendibilità. Per trovare un valore che permetta di distinguere correttamente tra soggetti genuini e impostori, definiamo:

$$S(\varphi(p), \bar{\varphi}) = \begin{cases} 1 - \bar{\varphi} & \text{if} \quad \varphi(p) > \bar{\varphi} \\ \bar{\varphi} & \text{otherwise} \end{cases}$$

L'indice SRR può infine essere definito come:

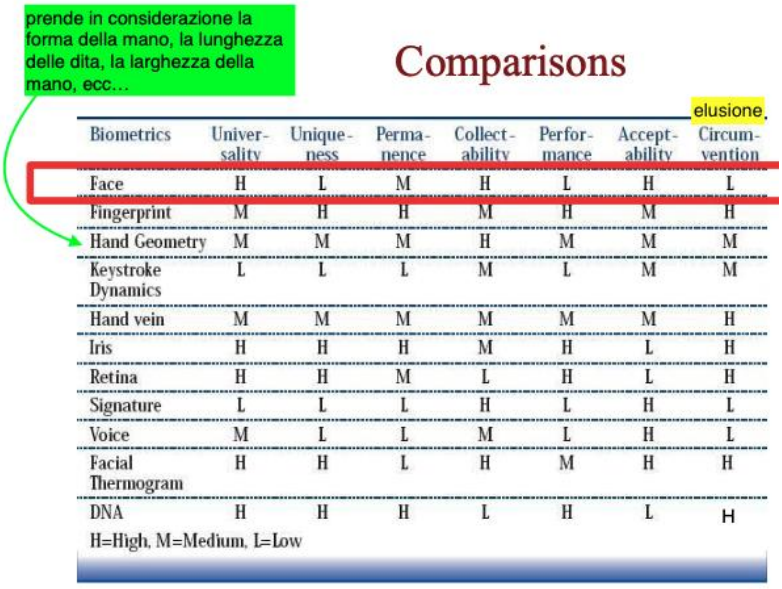
$$SRR = (\varphi(p) - \bar{\varphi}) / S(\bar{\varphi})$$

FACE RECOGNITION: INTRODUCTION AND FACE LOCALIZATION

Lezione 4

I fattori più importanti che influenzano la fattibilità di una biometria sono l'***accuratezza/affidabilità*** e l'***accettabilità***.
Uno dei sistemi più accurati è quello del ***riconoscimento del DNA*** ma allo stesso tempo è anche uno dei più invasi. Le ***impronte digitali*** sono accurate e accettate più facilmente, ma richiedono un

utente consapevole e collaborativo, molti possono associare il fatto di effettuare il riconoscimento delle impronte all'essere un criminale. Inoltre, in alcuni casi, potrebbero essere di bassa qualità (ad es., gran lavoratori che hanno le mani spaccate o persone che soffrono di una particolare sudorazione).
Il ***riconoscimento del volto*** invece è un sistema di autenticazione poco invasi e con un'alta accettabilità, ma la precisione deve migliorare, inoltre non è visto come un metodo invasivo di riconoscimento da parte degli utenti, poiché al giorno d'oggi si è molto abituati a farsi fotografare.



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

H=High, M=Medium, L=Low

I problemi più comuni sono:

- ***variazioni intrapersonali***, sono differenze che si osservano all'interno della stessa persona quando vengono valutate in momenti diversi o in situazioni diverse.
- ***somiglianze interpersonali***, sono differenze che si osservano tra le persone.
- ***Invecchiamento***, ***variazioni di posa***, ***illuminazione*** ed ***espressione*** (A-PIE, Pose Illumination Expression),
- ***eventuali travestimenti maligni*** (trucco, chirurgia plastica, occhiali, sciarpa...).

Alcuni Database molto famosi sono:

- ***FERET 1996***:
  - contiene variazioni di posa, illuminazione e tempo (solo la prima è controllata),
  - contiene un totale di 14051 immagini suddivise in diverse categorie, con posizione occhi e bocca per ogni immagine;
- ***AR-Faces 1998***:
  - le immagini dei volti hanno diverse espressioni facciali, condizioni di illuminazione, occlusioni e tempo (tutti
  - controllato),
  - contiene oltre 4000 immagini di 126 persone;
- ***CMU-TORTA 2000***:
  - contiene oltre 40.000 immagini da 68 soggetti,
  - ogni persona viene ripresa in 13 diverse pose, 43 diverse condizioni di illuminazione e con 4 diverse espressioni;
- ***CASIA 3D Volto V1 2001***:
  - 4624 scansioni di 123 persone utilizzando il digitalizzatore 3D senza contatto;

- ***LFW (Facce etichettate in natura) 1007***:
  - contiene più di 13.000 immagini di volti raccolte dal web,
  - ogni volto è stato etichettato con il nome della persona nella foto,
  - 1680 delle persone nella foto hanno due o più foto distinte nel set di dati
  - l'unico vincolo su questi volti è che sono stati rilevati dal rilevatore di volti Viola-Jones;
- ***YouTube faces 2007***:
  - contiene 3.425 video youtube di 1.595 persone diverse, – sono disponibili in media 2,15 video per ogni argomento,
  - la durata media di un video clip è di 181,3 fotogrammi.

MORE ABOUT FACE LOCALIZATION

Lezione 5

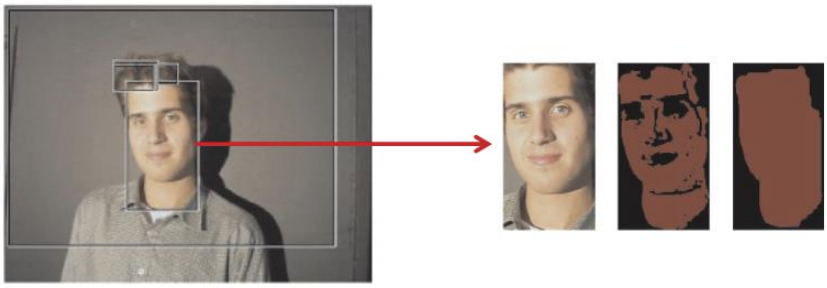
Problema: data una singola immagine o una sequenza video, rilevare la presenza di uno o più volti e individuare la loro posizione all'interno della singola immagine.
Requisiti: è necessario essere indipendenti rispetto a posizione, orientamento, scala, espressione (possibilmente diversa per i diversi soggetti dell'immagine), illuminazione, sfondo disordinato.

Esiste un algoritmo, per effettuare il face detection, chiamato "***Algorithm A by Hsu, Mottaleb and Jain, 2002***".
Tale algoritmo si basa su diverse fasi:

- ***Illumination compensation***: il tono della pelle dipende dall'illuminazione generale della scena, quindi per normalizzare i colori si usano riferimenti di bianco.
  Consideriamo i pixel con il 5% superiore della luminanza, valori come il bianco di riferimento se il numero di questi pixel bianchi di riferimento è maggiore di 100. I componenti rosso, verde e blu di un'immagine a colori vengono regolati in modo che questi pixel bianchi di riferimento vengono scalati al livello di grigio di 255.
- ***Color space transformation***: la modellazione del colore della pelle richiede la scelta di uno spazio colore appropriato e l'identificazione di un cluster associato al colore della pelle in questo spazio. Usiamo lo spazio $\mathbf{YC_bC_r}$ poiché è ampiamente utilizzato negli standard di compressione video. Poiché il colore della tonalità della pelle dipende dalla luminanza, trasformiamo in modo non lineare lo spazio colore $\mathbf{YC_bC_r}$ per rendere il luma del cluster di pelle indipendente. Ciò consente anche un rilevamento affidabile dei colori della tonalità della pelle scuri e chiari.
- ***Localization based on skin model***: abbiamo due alternative per effettuare questo step:
  - *Variance-Based Segmentation*: il metodo più semplice di segmentazione dell'immagine è il *thresholding method*. La chiave è selezionare il miglior valore di soglia, che può essere fatto con il metodo della ***massima entropia***, il ***metodo di Otsu*** (massima varianza) o il ***clustering k-means***. Per il metodo di *Otsu*, sia i un'immagine in scala di grigi con L livelli di grigio di dimensione $M \times N$, $f(x, y)$ il valore di grigio del pixel nel punto (x,y), n_i il numero di pixel con il grigio livello i, C_0 e C_1 le classi di livello di grigio prodotte da una soglia t (tipicamente oggetto e sfondo), allora la soglia ottimale è

$$t^* = \underset{t\in[0,\dots,L-1]}{\arg\min}\ (\sigma_w^2(t))$$

- *Connected Components*: i toni dei pixel della pelle sono iterativamente segmentati usando colori locali, raggruppando componenti in base a colori simili, fino a generare insiemi di candidati per il viso.



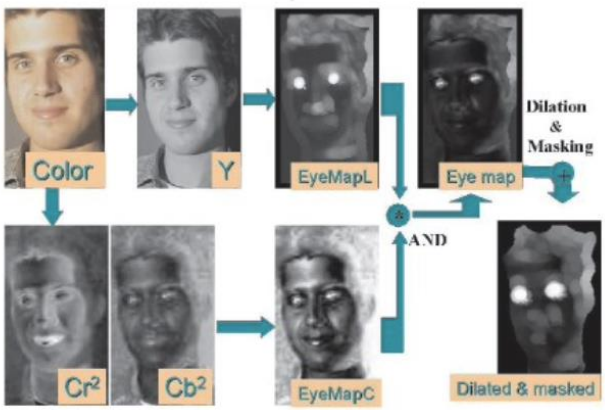
- ***Localization of main features (occhi, bocca,…)***:
  - **Localizzazione degli occhi**: l'algoritmo costruisce due mappe degli occhi, ovvero la ***Chrominance map*** e la ***Luminance map***.
    - ***Chrominance map***: si basa sull'osservazione che la regione intorno agli occhi è caratterizzata da alti valori di C_b e bassi valori di C_r.

$$EyeMapC = \frac{1}{3}\left\{\left(C_b^2\right)+\left(\tilde{C}_r^2\right)+\left(\frac{C_b}{C_r}\right)\right\} \qquad \tilde{C}_r = 255 - C_r$$

      Dove tutte le C sono normalizzate in un range di valori compresi tra 0 e 255.
    - ***Luminance map***: costruisce questa mappa, basandosi sulla consapevolezza che gli occhi solitamente contengono zone chiare e zone scure che possono essere evidenziate da operatori morfologi (*dilation* ed *erosion*).

$$EyeMapL = \frac{Y(x,y) \oplus g_\sigma(x,y)}{Y(x,y) \ominus g_\sigma(x,y) + 1}$$

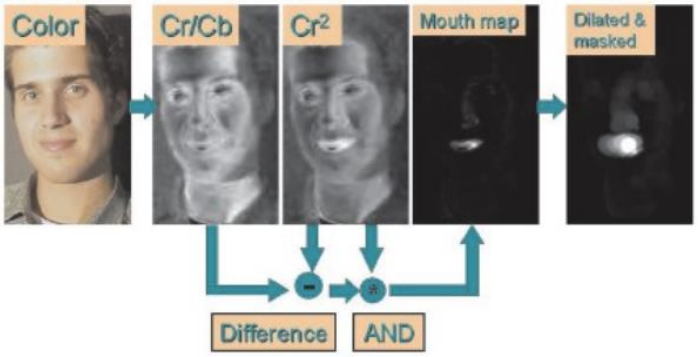
      Dove il + indica la *dilation* ovvero proviamo ad espandere certe zone per arrivare al nostro scopo, mentre il – indica l'*erosion* ovvero vado a diminuire l'immagine. Mentre g_σ indica lo *structuring elements*.



    Le due mappe vengono combinate tramite l'operatore AND. La mappa risultante subisce dilatazione, mascheramento e normalizzazione per scartare le altre regioni del viso e illuminare gli occhi.
  - **Localizzazione della bocca**: l'algoritmo costruisce la mappa della bocca basandosi sul fatto che, nella regione della bocca, la componente **C_r è <u>maggiore</u> di quella C_b**, e la risposta a **C_r/C_b** è **<u>bassa</u>**, mentre la risposta a **C_r^2** è **<u>alta</u>**

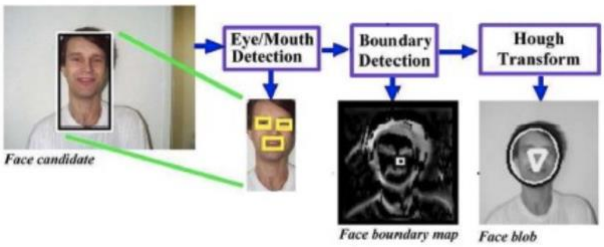
$$MouthMap = C_r^2 \cdot \left(C_r^2 - \eta \cdot \frac{C_r}{C_b}\right)^2 \quad \text{with } \eta = 0.95 \cdot \frac{\frac{1}{n}\sum_{(x,y)\in FG} C_r(x,y)^2}{\frac{1}{n}\sum_{(x,y)\in FG} \frac{C_r(x,y)}{C_b(x,y)}}$$

C_r/C_b e **C_r^2** sono normalizzati in un range che va da 0 a 255. *n* è il numero di pixel nella face mask.



- **Face contour**: l'algoritmo analizza tutti i triangoli composti da due occhi candidati e una bocca candidata. Ogni triangolo è verificato in questo modo:
  - variazioni di luma e media del gradiente di orientamento delle macchie contenenti occhi e bocca;
  - geometria e orientamento del triangolo,
  - presenza di un contorno del viso attorno al triangolo,

  Il triangolo con lo score più alto viene selezionato.



Un altro algoritmo per la localizzazione del volto è chiamato "***Algoritmo B di Viola e Jones***". Prevede di creare un classificatore che inizialmente è addestrato mediante multiple istanze della classe da individuare (esempi *positivi*), e varie istanze di esempi *negativi*, ovvero immagini che non contengono alcun oggetto della classe in esame.
Se il classificatore addestrato non trova un oggetto che invece è presente (***miss***) oppure ne indica erroneamente la presenza (***false alarm***), si può ricalibrare il suo addestramento aggiungendo gli esempi corrispondenti (positivi o negativi) al training set.
Usa un classificatore in grado di associare un pattern in input a una delle due classi ***volto/non volto***:

- Le immagini dei volti hanno proprietà comuni;
- Le immagini che non rappresentano volti sono estremamente irregolari.

Ci sono 3 contributi fondamentali all'algoritmo:

- Estrazione e valutazione di ***Haar-Like feature***
- Classificazione mediante ***Boosting***
- ***Multiscale detection***

La localizzazione è effettuata facendo scorrere una finestra di ricerca (le cui dimensioni possono variare) sull'immagine, estraendo le feature presenti nella finestra e classificando la finestra come volto o non volto.

L'bbiettivo dell'***AdaBoosting*** è quello di costruire un classificatore *non lineare complesso* $H_M(x)$ combinando M classificatori più semplici detti classificatori deboli (***weak classifiers***):

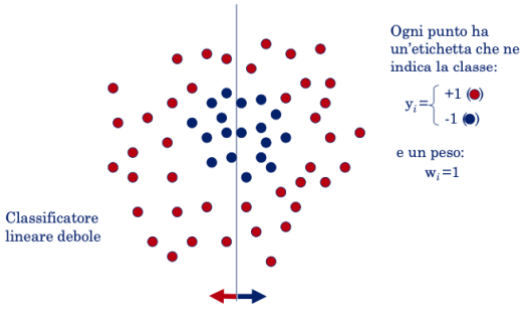
$$H_M(x) = \frac{\sum_{i=1}^{M} \alpha_i h_i(x)}{\sum_{i=1}^{M} \alpha_i}$$

dove:

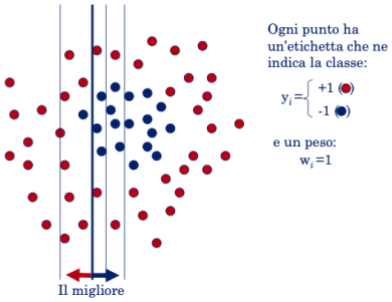
- x è il pattern da classificare
- $h_M(x) \in \{-1, +1\}$ sono i classificatori deboli
- $\alpha \geq 0$ sono i fattori di peso associati
- $\Sigma\alpha_m$ è il fattore di normalizzazione

AdaBoost è una tecnica di addestramento che ha lo scopo di apprendere la sequenza ottimale di classificatori deboli e i corrispondenti pesi.
Richiede un insieme di pattern di training $\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$, dove $y_i \in \{-1,+1\}$ è l'etichetta della classe associata al pattern. Inoltre durante l'apprendimento è calcolata e aggiornata una distribuzione di pesi $[w_1,w_2,\ldots,w_N]$ associati ai pattern di training, w_i è associato al pattern (x_i,y_i).
Dopo l'iterazione m, è assegnato ai pattern <u>*più difficili da classificare*</u> un peso $w_1(m)$ ***superiore***, cosicché alla successiva iterazione $m+1$ tali pattern riceveranno un'attenzione maggiore.
Esempio:

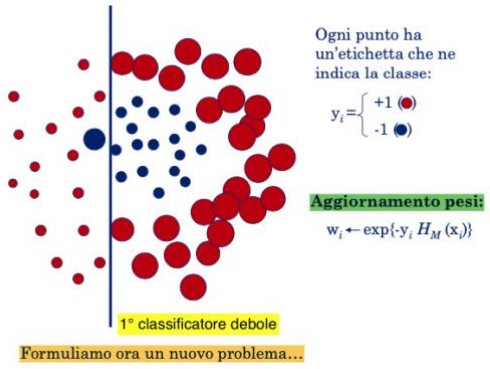
1)



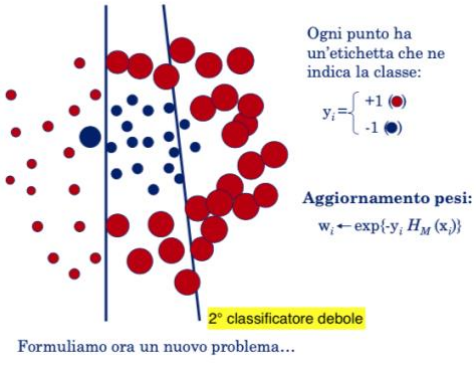
2)



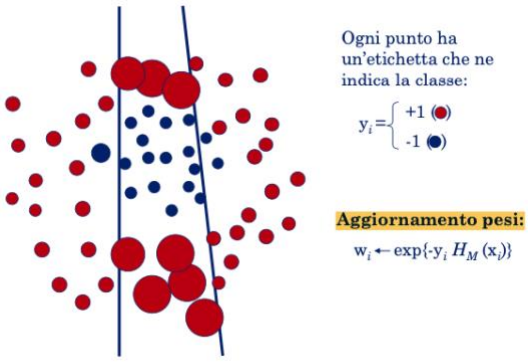
3)



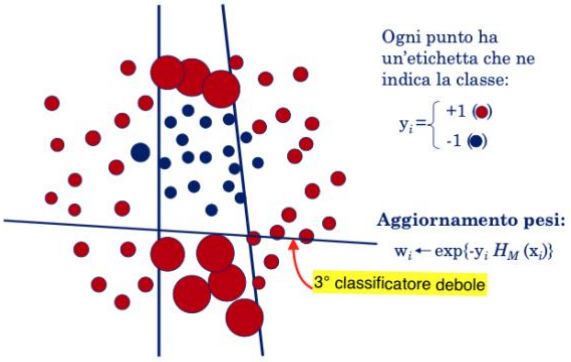
4)



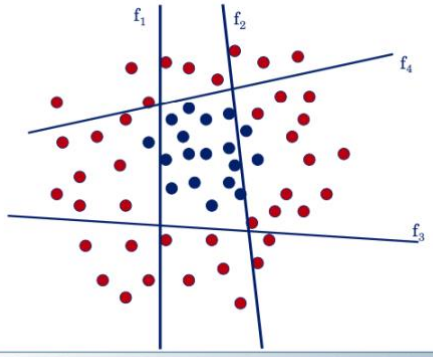
5)



6)

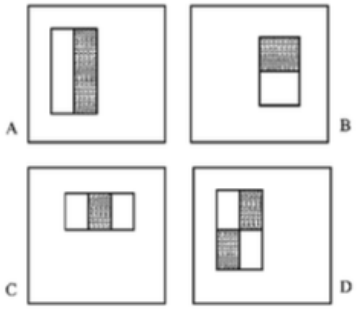


7)



Il classificatore robusto (**non lineare**) è costituito da una combinazione di classificatori deboli (**lineari**).

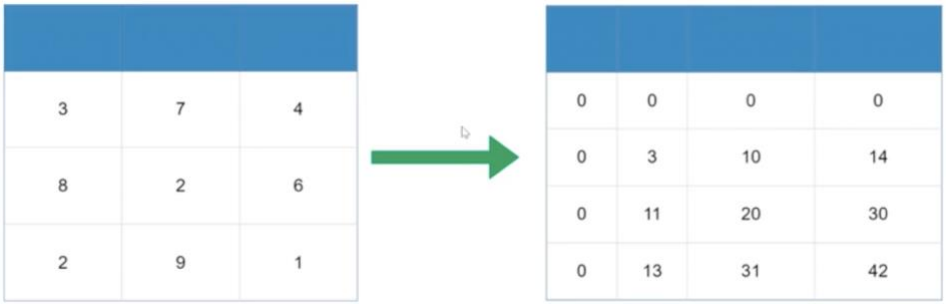
Per quanto riguarda le immagini, possiamo definire ***weak-learner*** in base a caratteristiche semplici (***rettangolari***): le ***Haar-like features***.



Quello che fa è calcolare: ***Σ(pixel nell'area bianca) - Σ(pixel nell'area nera)***. Se il risultato dell'operazione è un numero grande allora vuol dire che con alta probabilità in quellaporzione di immagine è presente la features identificata dal filtro (il filtro è uno dei quadrati sopra), dove ad esempio nel caso del B (nell'immagine sopra) sono angoli.
Per un immagine 24x24px, il numero di possibili rettangoli di features è 160'000!
Come si calcolano le Haar features?
Le rectangular features possono essere valutate attraverso ***immagini integral*** il quale nome viene dato, in ambito computer vision, ad un algoritmo con annessa struttura dati chiamata ***Summed-Area table***, la quale ci consente di calcolare l'area di una sottomatrice in tempo costante.
L'*immagine integrale* in posizione ***(x,y)*** è la somma del valore dei pixel sopra e a sinistra di ***(x,y)***:

$$II(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y')$$

dove $II(x,y)$ indica l'immagine integrale in ***(x,y)***.

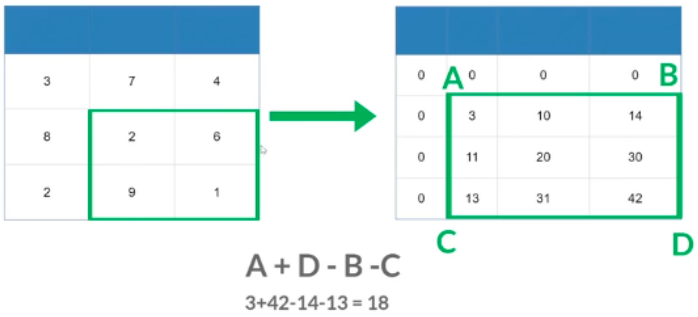


3	7	4
8	2	6
2	9	1

0	0	0	0
0	3	10	14
0	11	20	30
0	13	31	42

Si aggiunge una nuova riga ed una nuova colonna, rispettivamente sopra ed a sinistra, composte da soli 0. Per ogni valore nella matrice si va a calcolare la somma del valore presente in una cella con ogni valore ad esso sopra ed a sinistra. Ad esempio il ***9*** nella matrice a sinistra diventa nella nuova matrice 31 poiché è il risultato della seguente somma: *sinistra*(9+2) + *sopra*(2+7+3+8) = 31.

Usando l'immagine integrale, è possibile calcolare la somma del valore dei pixel in qualsiasi rettangolo, utilizzando solo quattro valori:



Prendiamo una parte della matrice di cui vogliamo conoscere l'area (o somma dei pixel) (quadrato verde a sinistra), poi prendiamo la corrispondente parte all'interno della summed-area table (quadrato verde a destra), la cui avrà la stessa posizione di origine della matrice originale a sinistra, ma un numero di colonne e righe maggiori di uno. Per conoscere l'area nella summed-area table dobbiamo usare la formula ***A + D - B – C***. Se facciamo la somma nel quadrato a sinistra, 2+6+9+1 il risultato è 18, se la calcoliamo nella summed-area table con la formula sopra, 4+42-14-13 il risultato è sempre 18.
Con la summed-area table il numero di operazioni per calcolare l'area è sempre costante, ovvero sempre quattro operazioni dobbiamo fare, indipendentemente dalla grandezza del quadrato, mentre nel caso a sinistra se scegliamo un quadrato di diemnsioni 6*6 dovremmo fare la somma tra 36 numeri differenti.

Per quanto riguarda invece il ***classificatore debole*** (***weak classifier***) abbiamo un albero di decisione con un singolo nodo.
Supponiamo di aver costruito ***M-1*** classificatori deboli ***{$h_m(x)|m=1,..,M-1$}*** e di voler costruire ***$h_M(x)$***. Il classificatore confronta il valore di una feature ***z_k**** con un treshold prefissato ***τ_k**** e assegna i valori **+1** o **-1** di conseguenza:

$$h_M(x) = +1 \text{ if } z_{k*} > t_{k*}$$
$$= -1 \text{ otherwise}$$

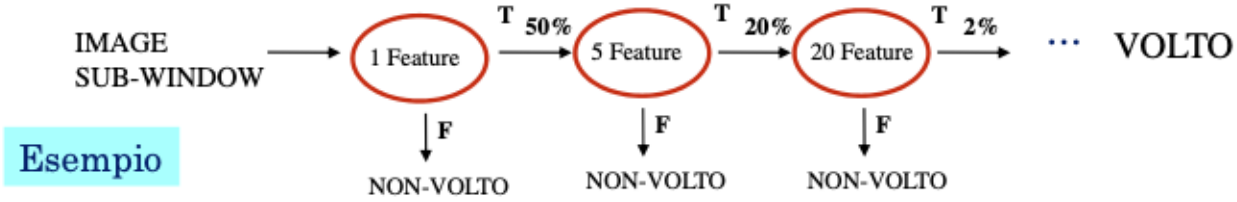
$h_M(x)$ dipende da due valori, ovvero dalla feature ***z_k**** e dal treshold ***τ_k****.

I due parametri (z_k^* e τ_k^*) possono essere fissati in base al minimo errore pesato di classificazione:

- Scegliamo il valore di soglia τ_k che minimizza l'errore di classificazione per ciascuna feature z_k;
- La caratteristica z_k^* scelta per il classificatore corrente è quella che permette di ottenere l'errore di classificazione complessivamente più basso.

AdaBoost apprende una sequenza di classificatori deboli h_m e li combina in un classificatore robusto H_M, minimizzando l'*upper bound* all'errore di classificazione di H_M.

Un solo classificatore robusto, per quanto elimini una grande porzione di sottofinestre che non contengono facce, non soddisfa i requisiti di applicazioni. Una possibile soluzione consiste nell'impiego di ***classificatori in cascata*** (***cascade classifier***), via via più complessi:



dove:

- un classificatore per ***1 sola feature*** riesce a passare al secondo stadio la quasi totalità dei volti esistenti (circa 100%) mentre scarta al contempo il 50% dei falsi volti.
- un classificatore per ***5 feature*** raggiunge quasi il 100% di detection rate e il 40% di false positive rate (20% cumulativo) usando i dati dello stadio precedente.
- un classificatore per ***20 feature*** raggiunge quasi il 100% di detection rate con 10% di false positive rate (2% cumulativo).

La localizzazione dei volti avviene analizzando sottofinestre consecutive (sovrapposte) dell'immagine in input e valutando per ciascuna se appartiene alla classe dei volti:

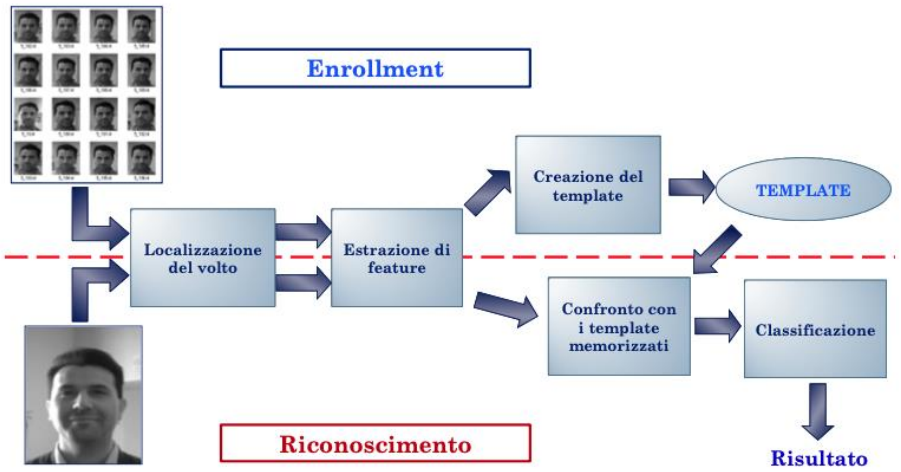
L'addestramento è molto lento (può richiedere giorni) ma la procedura di localizzazione è molto efficiente (funzionamento in real-time).

VALUTAZIONE DELLE PRESTAZIONI DI LOCALIZZAZIONE: INDICATORI

- ***Falsi positivi***
  - Percentuale di finestre classificate come volto che in realtà non lo contengono.
- ***Facce non localizzate***
  - Percentuale di volti che non sono stati individuati
- ***C-ERROR***
  - *Errore di localizzazione*: distanza euclidea tra il reale centro della faccia e quello ipotizzato dal sistema, normalizzato rispetto alla somma degli assi dell'ellisse contenente il volto.

FACE RECOGNITION – 2D

Lezione 6



Un'immagine può essere rappresentata come un insieme di punti in uno spazio multidimensionale. Una funzione ***RGB*** definita in uno spazio Cartesiano può assegnare un valore tra 0 e 255 a tutti i punti *(x, y)* nel piano.
Questi valori possono essere salvati in una matrice $\boldsymbol{w * h}$ o in un vettore unidimensionale $n = w * h$.
Più in generale, un'immagine *I* può essere rappresentata come un punto in uno spazio multidimensionale.
L'aspetto negativo della rappresentazione dell'immagine è la sua dimensione.

La rappresentazione di un'immagine tramite il valore dei suoi livelli di grigio dà luogo a un vettore di feature di dimensionalità molto elevata ($w \times h$). La gestione di questo tipo di dati è molto problematica a causa di un fenomeno noto come ***curse of dimensionality***:

- I dati sono spesso affetti da rumore;
- Alcune dimensioni (pixel) non portano informazioni significative ai fini del riconoscimento;
- I dati sono sparsi: è necessaria una grande quantità di immagini di training per avere una conoscenza adeguata dello spazio;
- Il confronto è un'operazione molto costosa in termini di complessità computazionale.

Metodi come il ***Principal Component Analysis*** (***PCA***) mirano a ridurre i set di dati ad alta dimensionalità a dati a dimensione inferiore senza una significativa perdita di informazioni, identificando le dimensioni più informative ottenendo sottospazi rilevanti.

La ***Principal Component Analysis*** (***PCA***) è una procedura statistica che utilizza una trasformazione ortogonale per convertire un insieme di osservazioni di variabili eventualmente correlate (è coinvolta una certa ridondanza) in un insieme di valori di variabili linearmente non correlate (nessuna ridondanza) chiamate componenti principali.
In questo modo, lo spazio delle caratteristiche n-dimensionali viene proiettato su un sottospazio ***k***-dimensionale.
Il valore di ***k*** ottimale viene calcolato a partire da un insieme di ***m*** campioni n-dimensionali che costituiscono il training set:

$$\boldsymbol{TS} = \{\mathbf{x}_i \in \mathfrak{R}^n | i=1,\ldots m\}$$

Dato il TS, possiamo calcolare la ***mean vector***:

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

Con la mean vector prendiamo gli *m* samples (i vettori n-dimesionali) sommiamo le componenti per poi andare a dividere questa somma per il numero di sample. In questo modo possiamo andare a catturare le caratteristiche comuni di tutte le immagini. In questo modo sembra che sia stato creato un nuovo volto. La parte più definita dell'immagine `e quella rappresentante gli occhi. Questo accade perch`e i soggetti, pur essendo notevolmente diversi tra loro, hanno una certa somiglianza in questa parte.

Questa è la mean face.

Riconsideriamo ora una singola immagine del dataset e sottraiamo ad essa la faccia media.

Ripetendo il procedimento per ogni faccia, al fine di concentrarci sulle caratteristiche che rendono particolari i vari soggetti, otteniamo un nuovo database:

Eigenfaces (eigenvector)

Per ripetere questo procedimento per tutti i componenti del dataset calcoliamo la ***matrice di covarianza*** per l'insieme di vettori TS:

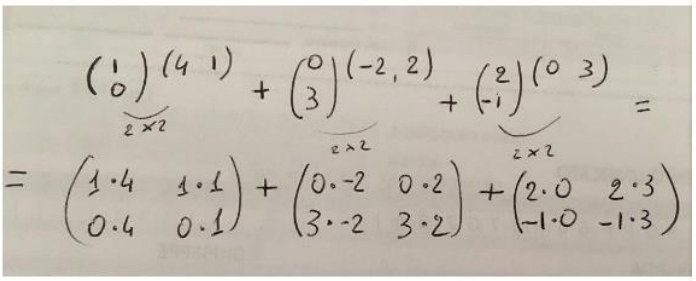
$$C = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})(x_i - \bar{x})^T$$

Prendiamo ogni valore x_i e lo sottraiamo al *mean vector* e lo moltiplichiamo per la trasposta della stessa sottrazione. Sottraggo il mean vector per ogni valore x_i poichè al suo interno (il mean vector) contiene informazioni ridondanti.
Esempio:

$$AB = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 3 & -1 \end{pmatrix}\begin{pmatrix} 4 & 1 \\ -2 & 2 \\ 0 & 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1\cdot 4 + 0\cdot(-2) + 2\cdot 0 & 1\cdot 1 + 0\cdot 2 + 2\cdot 3 \\ 0\cdot 4 + 3\cdot(-2) + (-1)\cdot 0 & 0\cdot 1 + 3\cdot 2 + (-1)\cdot 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 4+0+0 & 1+0+6 \\ 0-6+0 & 0+6-3 \end{pmatrix} = \begin{pmatrix} 4 & 7 \\ -6 & 3 \end{pmatrix}$$



La dimensione della matrice ***C*** di covarianza è ***n x n***. Il nuovo spazio ***k-dimensionale*** è definito dalla matrice di proiezione le cui colonne sono i k autovettori di C corrispondenti ai k autovalori più alti di C.
Sirovich e Kirby furono i primi ad utilizzare PCA per il face recognition.
Loro hanno dimostrato tutte le particolari facce possono essere efficentemente rappresentate lungo l'***eigenpictures coordinate space***, e può essere ricostruito utilizzando una piccola collezione di ***eigenpictures*** e le corrispondenti proiezioni ("coefficienti") lungo ciascuna eigenpictures.

Le proiezioni vengono eseguite moltiplicando la ***matrice di proiezione trasposta*** per il ***vettore originale*** a cui è stata prima sottratta la sua ***media***:

$$Proj(x) = \varphi_k{}^T(x - \bar{x})$$

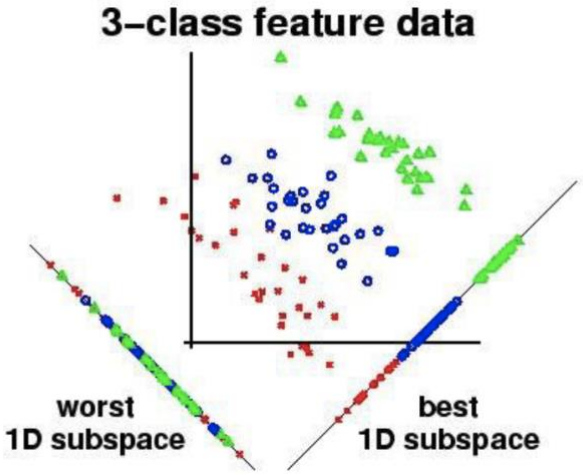
Le eigenfaces (o eigenvectors) definiscono uno spazio delle caratteristiche che riduce drasticamente la dimensionalità dello spazio originale, e l'identificazione delle facce viene effettuata in questo nuovo sottospazio ***KL***. Ogni immagine della galleria viene proiettata nel sottospazio ***KL*** ed i coefficienti di proiezione costituiscono il vettore delle caratteristiche.
In sostanza, per capire se un nuovo soggetto è presente nel database iniziale consideriamo la sua proiezione nel nuovo spazio e calcoliamo la distanza rispetto alle altre facce. Nel caso sia abbastanza vicino ad una di esse avviene il riconoscimento.

Anche la ***PCA*** presenta alcuni *problemi*:

- è buono per la rappresentazione, ma manca di un vero potere discriminatorio,
- quando ci sono variazioni significative di PIE (Pose Illumination Expression), esse possono interferire nella determinazione dell'identità degli individui, così le classi possono non essere separate correttamente;

Per risolvere i problemi di cui sopra Belhumer ha proposto il ***Fisherfaces***, un'applicazione del ***discriminante lineare di Fisher*** (***FLD***), più spesso menzionato nel contesto dell'***Analisi Discriminante Lineare*** (***LDA***).

LDA è una tecnica di riduzione di dimensionalità lineare e supervisionata il cui obiettivo è massimizzare la separazione tra le classi e mantenere le informazioni discriminatorie il più possibile. La *trasformazione dello spazio* è determinata sulla base di un criterio di ottimizzazione che ha lo scopo di *massimizzare la separazione tra le classi nello spazio ridotto*:



Un possibile approccio è il seguente:

- Partiamo da un insieme di ***m*** campioni *n-dimensionali* che costituiscono il *training set* ***T S = {xi ∈ R^n : i = 1,...,m}*** dove ***m*** è la cardinalità del TS.
- Diversamente da ***PCA***, il training set è partizionato in base alle classi.
- Cerchiamo di ottenere uno scalare ***y*** proiettando i campioni ***x*** su una retta tale che ***y = $w^T x$***
- Considero due classi ***P_1*** e ***P_2*** con vettori ***m_1*** e ***m_2***.
- Consideriamo i *mean vector* di ciascuna classe nei due spazi ***(μ con _ è il mean vector nel nuovo spazio, ovvero la retta***):

$$\mu_i = \frac{1}{m_i}\sum_{j=1}^{m_i} x_j$$

and

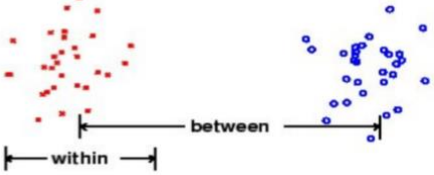
$$\tilde{\mu}_i = \frac{1}{m_i}\sum_{j=1}^{m_i} y_j = \frac{1}{m_i}\sum_{j=1}^{m_i} w^T x_j = w^T \mu_i$$

  Il mean vector nel nuovo spazio può essere anche rappresentato come il mean vector della proiezione del vettore iniziale sul nuovo spazio (ovvero di x_j che sta sopra d lui)
- Adesso vogliamo massimizzare la distanza tra i due *mean vector delle due classi*:

$$J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| \left| w^T(\mu_1 - \mu_2) \right|$$

  mu barretta 1 **meno** ***mu barretta 2*** non sono però delle buone misure perché non tengono in considerazione la standard deviation (non ci informa su quanto i dati varino) tra le classi per questo motivo si usa la SCATTERING MATRICES

Un approccio migliore consiste nel massimizzare il rapporto tra la varianza tra le classi e la varianza all'interno della classe a partire dalle matrici di scattering (separazione):



- Si parte da un ***Training Set*** (***TS***)
- Diversamente da ***PCA***, il training set è partizionato in base alle classi
- Per ogni classe P_i, calcoliamo il ***mean vector*** (un «centroide» per ogni classe in modo simile a PCA sull'intero insieme) e la media delle medie («centroide» di «centroidi»).

$$\mu_i = \frac{1}{m_i}\sum_{j=1}^{m_i} x_j \qquad \mu_{TS} = \frac{1}{m}\sum_{i=1}^{s} m_i \mu_i$$

- Calcoliamo la ***matrice di covarianza*** per ogni calsse P_i (in modo simile a PCA sull'intero insieme):

$$\mathrm{C_i} = \frac{1}{m_i}\sum_{j=1}^{m_i}(x_j - \mu_i)(x_j - \mu_i)^T$$

- Ora possiamo calcolare S_w ed S_b:

$$S_W = \sum_{i=1}^{s} m_i C_i$$

$$S_B = \sum_{i=1}^{s} m_i (\mu_i - \mu_{TS})(\mu_i - \mu_{TS})^T$$

m è il numero di elementi collezionati per ogni classe

S_w è dato dalla somma pesata della matrice di covarianza.

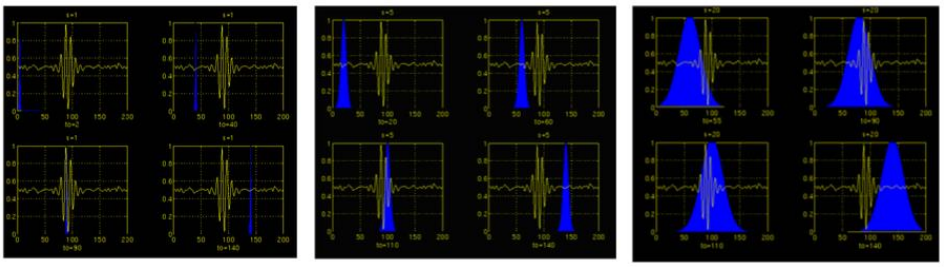
Fisher solution:
N è il numero di immagini nel database, c il numero di persone nel db. Ogni volto è rappresentato da un vettore, si calcola la media di volti per persona, si sottrae la media alla Training Faces, si calcola la Scatter Matrix e si massimizza il rapporto tra S_W e S_B.

FEATURE SPACE

Le caratteristiche salienti dell'immagine del volto possono essere individuate applicando all'immagine stessa filtri, trasformate, operatori, ciascuno progettato per mettere in luce particolari proprietà.
I vettori di feature estratti possono eventualmente essere sottoposti a un processo di riduzione della dimensionalità usando una delle tecniche precedentemente descritte.

WAVALET TRANFORMS
Il ***Fourier Transform*** (***FT***) fornisce le informazioni sulla frequenza del segnale (in questo caso le immagini), il che significa che ci dice quanto di ciascuna frequenza esiste nel segnale, ma non ci dice quando nel tempo esistono queste componenti di frequenza.
Gli spettri di due segnali completamente diversi possono sembrare molto simili, poiché il FT fornisce il contenuto spettrale del segnale, ma non fornisce informazioni su dove nel tempo compaiono tali componenti spettrali. Pertanto, FT può essere utilizzato per segnali non stazionari, se siamo interessati solo a quali componenti spettrali esistono nel segnale, ma non siamo interessati a dove si verificano.

Mentre la ***Wavalet Transforms*** (***WS***) è in grado di fornire simultaneamente le informazioni di tempo e frequenza, fornendo quindi una rappresentazione tempo-frequenza del segnale.



La linea gialla è il segnale originale, mentre quella blu è la wavelet

Viene fatta passare una *Onda Madre* (quella in blu) nel dominio del tempo (linea gialla) con diverse dimensioni. Maggiore sarà la dimensione dell'onda madre maggiori saranno le feature catturate e discorso analogo vale per la dimensione minore della Onda Madre.
Esempio: Da un'immagine 2D vogliamo conoscere i livelli di grigio e inoltre dove questi livelli appaiono nell'immagine.
Gli algoritmi Wavelet elaborano i dati a diverse scale o risoluzioni. Se osserviamo un segnale con una "grande finestra" (grande onda blu), noteremmo caratteristiche grossolane. Allo stesso modo, se osserviamo un segnale con una "piccola finestra" (piccola onda blu), noteremmo piccole caratteristiche.

GABOR FILTER

Il *Gabor Filter* è un filtro lineare per rilevare i bordi di un'immagine. Si pensa che l'analisi delle immagini con i filtri Gabor sia simile alla percezione nel sistema visivo umano.
Si basa su una funzione Gaussiana:

$$\psi_{f,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta_n}^2}{\sigma_x^2}+\frac{y_{\theta_n}^2}{\sigma_x^2}\right\}\right]\exp\left(i\left(2\pi f x_{\theta_n}\right)\right)$$

La matrice di trasformazione viene moltiplicata con il vettore originale in modo tale da calcolare la risposta:

$$\text{where, } \begin{bmatrix} x_{\theta_n} \\ y_{\theta_n} \end{bmatrix} = \begin{bmatrix} \sin\theta_n & \cos\theta_n \\ -\cos\theta_n & \sin\theta_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

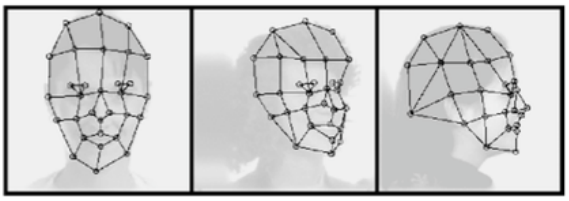
M. TRASFORMAZIONE

I feature vector vengono ottenuti facendo la convoluzione dell'immagine con una serie di gabor filter. Se si facesse la convoluzione pixel per pixel la dimensionalità sarebbe troppo elevata per ovviare a questo problema si possono applicare due soluzioni:

1. Applicare una griglia fissata di punti inserita a mano
2. Considerare solo le regioni con il maggior di informazione

Un'applicazione del filtro Gabor è ***Elastic Bunch Graph Matching*** (***EBGM***):

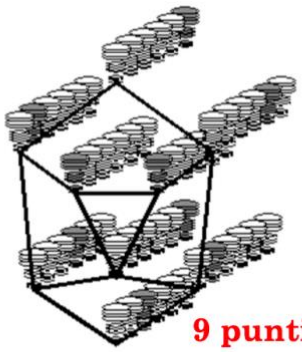
- La rappresentazione dell'oggetto di base è un grafico etichettato, definito come grafico a grappolo, uno per ogni posa, che raccoglie informazioni specifiche della classe;
- I bordi sono etichettati con le informazioni sulla distanza ed i nodi sono etichettati con le risposte *Wavelet* di *Gabor* raccolte localmente nei *jet*, ovvero una rappresentazione compatta e robusta di una distribuzione locale di livelli di grigio;



Esempio di grafo adattato per un volto in pose diverse.

La geometria di un volto è codificata dagli archi, mentre la distribuzione dei valori di grigio viene codificata dai nodi. Per descrivere una vista bidimensionale è sufficiente un solo grafo, mentre per rappresentare più viste o per ottenere una vista 3D occorre integrare più grafi.

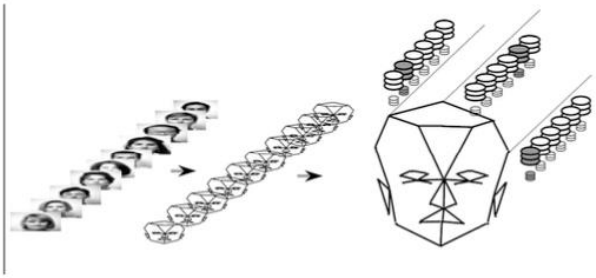
Un ***Face Bunch Graph*** (***FBG***) può rappresentare volti in generale. È progettato per coprire molteplici variazioni nell'aspetto dei volti. Combina informazioni di un certo numero di grafi di volto. I nodi sono etichettati con insiemi di jet detti ***bunches***, e gli archi sono etichettati con medie di vettori di distanza. Durante il confronto di un'immagine, si seleziona il jet più appropriato in ogni bunch, colorato di griglio in figura:



9 punti cardine

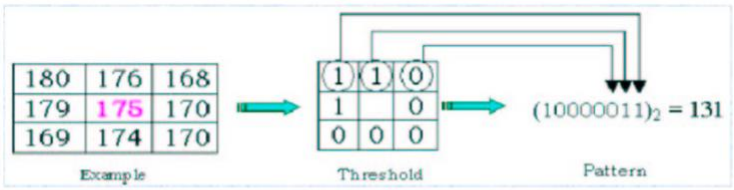
Benché costruito con solo sei volti campione, questo FBG è potenzialmente in grado di rappresentare (6^9= 10'077'696) volti diversi.

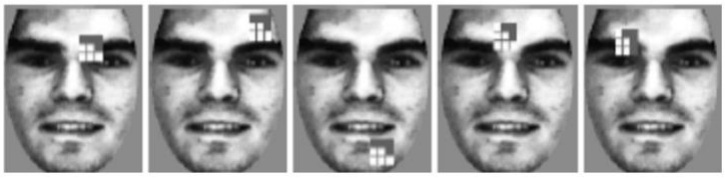
Un FBG è costruito a partire da un insieme rappresentativo di grafi modello (es. 70) aventi la stessa posa (es. frontale, di profilo, ...), dunque la stessa struttura. Ogni nodo è etichettato con tutti i jet presi dai modelli nei medesimi punti. Nuove facce possono essere rappresentate prendendo jet da differenti modelli, ad esempio occhio sinistro da un modello e naso da un altro modello.



LOCAL BINARY PATTERN

L'operatore assegna ai pixel di un'immagine, in un intorno di *dimensione 3×3*, un valore binario (0 o 1).
Sia ***p*** un pixel dell'intorno e p_c il pixel *centrale*, il valore binario è assegnato confrontando il valore del pixel ***p*** con quello del pixel p_c : se ***p*** ha un valore superiore o uguale a quello di p_c allora a ***p*** è assegnato il valore 1, altrimenti il valore 0. Questa operazione viene chiamata *tresholding*





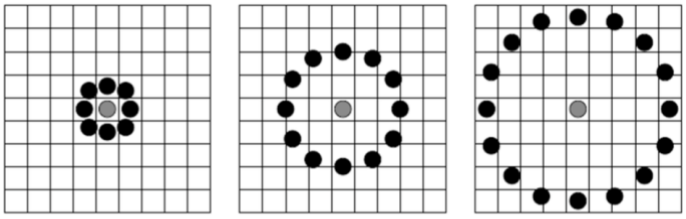
131 sarà il nuovo valore di p_c

Immagine ottenuta sostituendo i valori dei pixel originali con i valori binari ottenuti da LBP

L'istogramma viene utilizzato come *feature vector*, infatti possiamo confrontare due immagini considerando gli istogrammi.

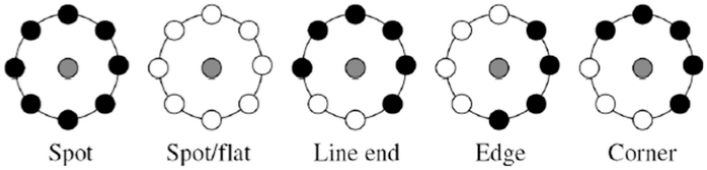
L'operatore LBP di base è stato esteso per gestire intorni di dimensione variabile di un pixel.
L'operatore in questo caso è definito da due parametri:

- il numero di punti campione **P**;
- il raggio dell'intorno circolare **R**.



Più sono i vicini e più dettagli si ottengono.

I pattern binari più interessanti sono quelli ***uniformi***, in quanto rappresentano le strutture locali più rilevanti (es. edge, spot, ecc).



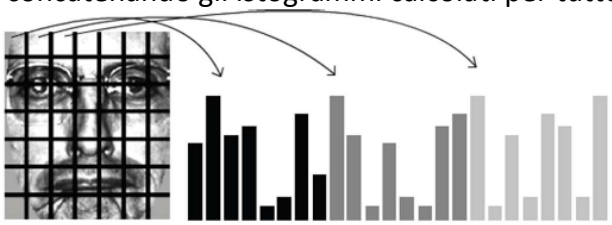
Un pattern è detto ***uniforme*** quando, considerato in modo circolare, contiene al massimo due transizioni 0-1 o 1-0. Ad esempio, i pattern 10000011, 11110000, 00000000 sono *uniformi*.

Considerare solo i pattern uniformi permette di risparmiare memoria: i pattern totali sono 2^P, mentre i pattern uniformi sono solo *P x (P-1)+2*.

Il *feature vector* associato a un'immagine è un ***istogramma*** calcolato come segue:

- L'immagine è partizionata in *k x k* sotto-finestre.
- Per ogni sottofinestra è costruito un istogramma in cui ciascun bin è associato a uno schema specifico; i bin sono in totale Px(P-1)+3:
    - Px(P-1) bin per i pattern con 2 transizioni,
    - 2 bin per i pattern con 0 transizioni,
    - 1 bin per i pattern non uniformi.

Il *feature vector* si ottiene concatenando gli istogrammi calcolati per tutte le sotto-finestre.



Esistono anche le ***Rotation*** ovvero in pratica ci sono i pattern, che sono tipo "bordo", "angolo", ecc…
Per riconoscerli senza tenere conto dell'orientazione, puoi fare delle rotazioni sulla stringa binaria e prendere la stringa che ha il maggior numero di **0** nella parte dei bit più significativi (a sinistra)
Ottenuta questa stringa fai il confronto con le stringhe di ogni tipo di pattern, che possono essere codificate in modo più efficiente visto che si considerano solo quelle particolari rotazioni delle stringhe

UNA POSSIBILE CLASSIFICAZIONE DEI SISTEMI DI RICONOSCIMENTO

- Global Appearance Method: Si basano sull'intera immagine quindi non perdono alcuna informazione a discapito però del costo computazionale. Molti algoritmi sono flessibili e quindi possono essere modificati in modo tale da gestire la PIE. Richiedono però una grande relazione tra il training set e il test set.
    - vantaggi:
        - non distruggono alcuna informazione nell'immagine concentrandosi solo su regioni o punti di interesse limitati,
        - molti di questi algoritmi sono stati modificati per compensare variazioni e dimensioni di PIE;

  - Svantaggi:
    - la maggior parte di questi approcci parte dal presupposto di base che tutti i pixel dell'immagine sono ugualmente importanti,
    - queste tecniche sono computazionalmente costose,
    - richiedono un alto grado di correlazione tra il test e le immagini di allenamento,
    - non si comportano efficacemente con grandi variazioni di PIE;
- ***Local or Feature-Based Methods***: basati sulla rilevazione di punti o caratteristiche locali, robusti rispetto alla variazione dell'input ma manca di abilità discriminatoria:
  - Vantaggi:
    - consentono una rappresentazione compatta delle immagini del volto e un matching ad alta velocità;
  - Svantaggi:
    - l'implementatore di una di queste tecniche deve prendere decisioni arbitrarie su quali caratteristiche sono importanti,
    - se il set di funzionalità non ha capacità di discriminazione, nessuna quantità di elaborazione successiva può compensare quella carenza intrinseca;
- ***Neural Networks***: una rete neurale mira a simulare il modo in cui funzionano i neuroni del cervello. Ogni neurone è rappresentato da una funzione matematica basata sulle probabilità. Per riconoscere i volti, una scelta ottimale potrebbe essere quella di utilizzare un neurone per ogni pixel, ma questo approccio utilizza troppi neuroni. La soluzione è utilizzare una rete neurale per riassumere l'immagine in un vettore più piccolo e una seconda rete esegue il riconoscimento vero e proprio.
  - Vantaggi:
    - Questo approccio riduce l'ambiguità tra soggetti appartenenti a classi simili.
    - Con opportuni accorgimenti sono resistenti alle occlusioni.
  - Svantaggi:
    - Richiedono più di un'immagine per l'addestramento.
    - Alcune reti sono soggette a una serie di problemi, come l'overfitting e l'overtraining.
    - Quando il numero dei soggetti aumenta, diventano inefficienti (dimensione del database).

FACE RECOGNITION – 3D

Lezione 6

La versione 3D presenta meno problemi di immagine, ma nessuna rappresentazione del volto è sufficientemente robusta per tutti i tipi di variazioni.

APIE (Aging, Pose, Illumination and Expression) e make-up o accessori influiscono negativamente sul riconoscimento 2D, il 3D solo da Aging, Expression e accessori.

Variation	2D	3D
Pose	Affected 🔴	NOT Affected 🟢
Illumination	Affected 🔴	NOT Affected 🟢
Expression	Affected 🔴	Affected 🔴
Ageing	Affected 🔴	Affected 🔴
Makeup	Affected 🔴	NOT Affected 🟢
Plastic surgery	Affected 🔴	NOT Affected 🟢
Glasses, scarfes, etc.	Affected 🔴	Affected 🔴

Vantaggi 3D:

- maggiori informazioni,
- robustezza ad alcune distorsioni,
- possibilità di sintetizzare immagini 2D (approssimative) da pose ed espressioni 3D virtuali calcolate da un modello 3D.

Svantaggi 3D:

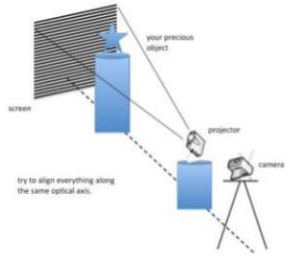
- costo dei dispositivi,
- costo computazionale delle procedure,
- possibile rischio (il laser scanner può essere pericoloso per gli occhi).

Rappresentazione:

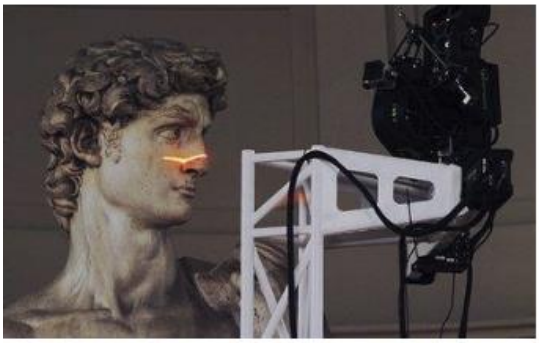
- **2D** il valore di ogni pixel è dato dall'illuminazione.
- **2.5D** è dato dalla distanza tra quel punto e la sorgente di luce ed è espresso solitamente in scala di grigi.
- **3D** la struttura è rappresentata come un insieme di punti e poligoni connessi in uno spazio 3D.

L'acquisizione delle immagini avviene in diversi modi:

- ***Camera Stereoscopica***: ha due o più lenti con un frame d'immagine per ogni lente, permette di simulare la visione binoculare umana.
- ***Structured light scanner***: viene proiettato sul soggetto un determinato pattern. L'informazione riguardante il modello 3D viene quindi catturato a seconda di come questo pattern viene deformato sul viso del soggetto.



- **Laser scanner**: Un unico raggio laser viene proiettato lungo la superficie (faccia). Il raggio viene deformato dalla struttura 3D della faccia e la deformazione dà la misura della profondità per ogni punto. La cattura deve essere ripetuta da diversi punti di vista per ottenere una modello 3D completo. Costo medio-alto, alta precisione, elevata robustezza all'illuminazione, 6-30 secondi per scansione, PERICOLOSO PER GLI OCCHI.



Le acquisizioni 3D presentano alcuni problemi:

- **Noise removal**: rimuovere i picchi utilizzando i filtri ed il rumore con i filtri medi;

- **Holes filling**: Smussamento gaussiano, interpolazione lineare, interpolazione simmetrica, operatori morfologici;
- **Smoothing + alignment**

La costruzione di un modello 3D avviene mediante l'integrazione di diversi modelli 2.5D i quali sono catturati da diverse angolazioni. Per ogni immagine 2.5D viene generata una nuvola di punti 3D, con coordinate x e y equidistanti e coordinate z (profondità) derivate dal valore nell'immagine 2.5D. Alla fine del processo questi punti vengono triangolarizzati in modo tale da ottenere il modello finale.

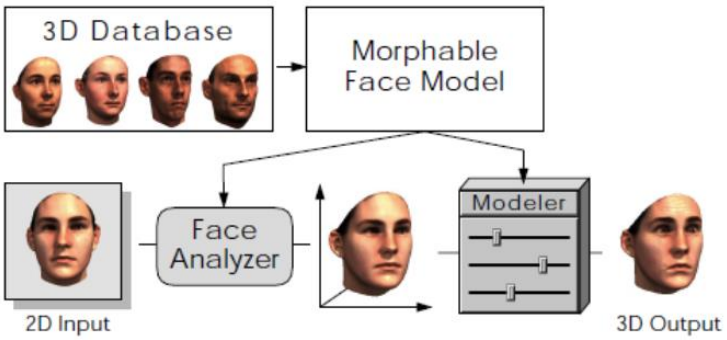
DALLA RAPPRESENTAZIONE 2D A QUELLA 3D: SHAPE FROM SHADING

Calcola una forma 3D dall'illuminazione del viso, assunta come *Lambertiana* (che riflette la luce in tutte le direzioni). Si può calcolare unendo forme 3D e applicando il PCA o usando soltanto l'immagine 2D in input usandola come guida.

MORPHABLE MODELS

A partire da un modello 3D generico iniziale si ottiene un altro modello 3D finale. I *morphable model* sono dunque rappresentazioni dinamiche della superficie del volto, che contengono informazioni sulla struttura dinamica (ad esempio muscolatura facciale) in aggiunta alle informazioni sulla geometria. Generazione del modello 3D finale:

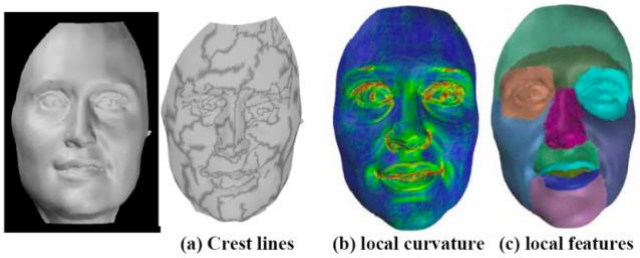
1. Si acquisiscono due o tre immagini fotografiche del volto (frontale, laterale, diagonale).
2. Si cambiano la forma e i colori di un modello generico di volto (morphable model) in accordo al contenuto delle immagini.
3. Si genera la texture combinando le due o tre immagini.



3D FEATURES

Gli algoritmi di riconoscimento facciale 3D lavorano spesso sulla curvatura locale e globale del modello facciale. È possibile estrarre le informazioni riguardanti la forma di una faccia 3D analizzando la curvatura locale della superficie. Esempi:

- **Crest Lines**: selezione delle aree con la maggiore curvatura
- **Local Curvature**: rappresenta la curvatura locale con un colore
- **Caratteristiche locali**: segmentazione del viso in regioni di interesse



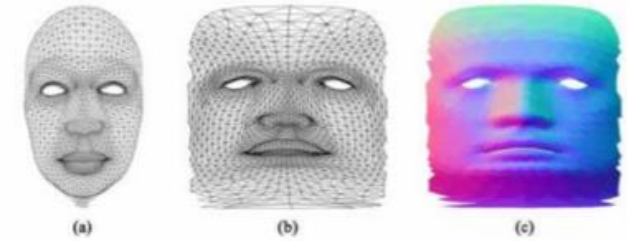
Prima del riconoscimento è importante l'allineamento che può essere fatto:

- COARSE: allineamento tra punti di riferimento
  - vengono trovati un numero finito di punti caratteristici
  - il viso viene allineato minimizzando la distanza tra i punti corrispondenti
- ICP (Iterative Closest Point): allineamento tra modelli
  1. Si trova un match iniziale tra le due superfici
  2. Si calcola la distanza tra le due superfici
  3. Si calcola la trasformazione che minimizza tale distanza
  4. Applica la trasformazione e reitera la procedura fin quando la distanza non sia sotto un certo threshold

3D FACE RECOGNITION

Bisogna estrarre le caratteristiche e fare l'allineamento. Diversi tipi di riconoscimento:

- **MAPPA NORMALE**
  a) acquisizione modello 3D
  b) proiezione da 3D a 2D
  c) generazione della mappa normale (immagine RGB)



Utilizzando la mappa normale otteniamo una rappresentazione bidimensionale dell'informazione tridimensionale. L'informazione sulla curvatura di un modello è rappresentata dall'insieme delle normali alla superficie. La lettura e l'elaborazione di un'immagine 2D sono molto più veloci rispetto alla lettura e all'elaborazione di un modello 3D.

- **FACEGEN MODELLER**: Tool che permette tramite i morphable model di creare dei modelli 3D partendo da un'immagine. È possibile quindi emulare le espressioni facciali.
- **ISO-GEODESIC STRIPES**: La distanza Geodetica è il percorso più breve tra due punti in uno spazio curvo, tali distanze sono influenzate dai cambiamenti delle espressioni. Le informazioni sono codificate in una rappresentazione sotto forma di grafo e il riconoscimento facciale è ridotto alla corrispondenza dei grafi. L'immagine del viso 3D è divisa in strisce facciali iso-geodetiche di uguale larghezza e distanza crescente dalla punta del naso.
  Ogni stripe può essere rappresentata come un nodo di un grafo. Due nodi sono connessi tra di loro tramite 3D WeightedWalkthroughts (3DWW).
  La similarità tra i due modelli avviene confrontando la distanza tra i nodi appartenenti alle stesse stripes dei due grafi.

FACE RECOGNITION EVALUATION

Lezione 8

Il riconoscimento facciale si tratta di un caso specifico di riconoscimento di forme, ma reso particolarmente difficile da fattori che possono influenzare l'aspetto del viso, come:

- ***Illumination***: il vettore associato a un soggetto in condizioni di illuminazione diverse può risultare più vicino a quello di un soggetto diverso con un'illuminazione simile. Si cerca di risolvere il problema in tre modi differenti:
  - ***Shape from shading***
  - ***Metodi basati sulla rappresentazione***: utilizzare classificatori che per loro natura intrinseca sono robusti rispetto alle variazioni di illuminazione.
  - ***Metodi Generativi***: a partire da un modello di volto 3D generano un ampio set di immagini con il maggior numero possibile di variazioni di illuminazione, che vengono poi utilizzate per l'arruolamento del soggetto;
- ***Pose***: in molte applicazioni la posa del soggetto durante il test può essere diversa da quella durante l'iscrizione. Per risolvere il problema ci sono due modi:
  - ***Muilti-view Systems***: registrare foto del soggetto in diverse pose
  - ***Pose correction systems***: partendo dall'immagine di prova il sistema deduce la posa del soggetto, cercando di correggerlo
- ***Occlusion***: esistono diversi tipi di occlusioni che nascondono parzialmente il viso, come occhiali da sole, mascherine, ecc… Per risolvere il problema si trova l'occlusione, si applica una *patch* in modo tale da dire al sistema che quella parte non è d'interesse.
- ***Time and age***: Le variazioni di tempo tra due immagini di uno stesso soggetto,
- possono ridurre le prestazioni di un sistema di riconoscimento.
- ***Make-up ed accessori***:
- ***Chirurgia estetica***:

FACE ANTISPOOFING

Lezione 8bis

Nel contesto della network security, un attacco di ***spoofing*** è una situazione in cui una persona o un programma si maschera con successo da un altro, falsificando i dati (indirizzo IP, indirizzo e-mail, ecc.) ottenendo così un vantaggio illegittimo.
Lo ***spoofing biometrico*** è l'atto di ingannare un'applicazione biometrica, utilizzando una copia o eseguendo un'imitazione del fattore biometrico che identifica il soggetto legittimo. Questo attacco viene effettuato presentando al sistema un tratto biometrico di un artefatto per ingannarlo fingendo di essere un utente autentico. Mentre nel ***camouflage*** Il soggetto che attacca il sistema si camuffa cercando di non essere riconosciuto.

Spoofing

Camouflage

FACE SPOOFING

Abbiamo face spoofing per immagini 2D e 3D:

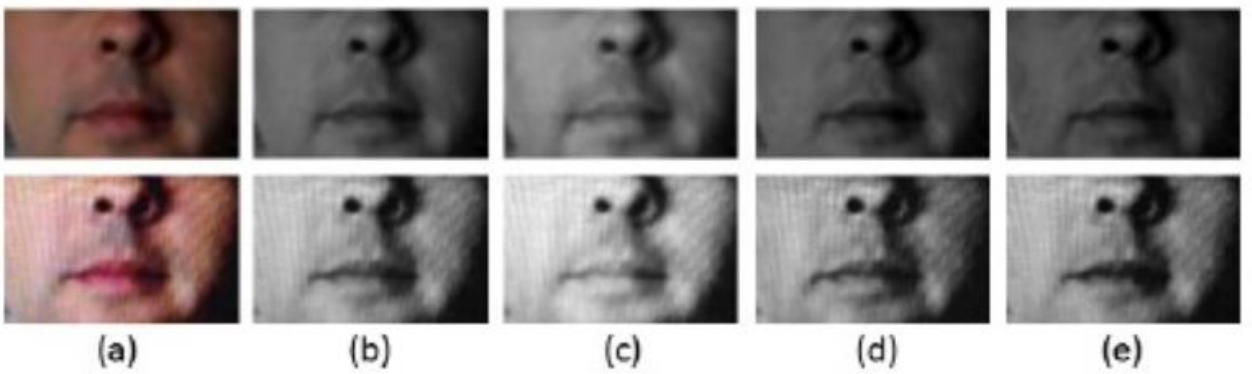
- 2D face spoofing: si presenta una foto o un video di uno sconosciuto.
- Spoofing facciale 3D: tramite delle maschere, andando a scansionare il volto da copiare e creando una vera e propria maschera.
- Operazioni chirurgiche

Possibili metodi di ***liveness detection*** per 2D face spoofing nelle ***foto*** sono:

- Movimento della testa
- Fondendo faccia-voce per verificare il movimento delle labbra
- Analisi del battito degli occhi, quindi si analizza la rapida chiusura ed apertura degli occhi che tipicamente accade ogni 3-4 secondi. È possibile definire un insieme di tre stati per gli occhi, ***Q = {α : aperto, γ : chiuso, β : ambiguo}.*** Una tipica attività di battito di ciglia può essere descritta come un modello di cambiamento di stato di **α → β → γ → β →α**.
- Le *foto* contengono un materiale (*metallo*) che riflette in maniera diversa la luce rispetto ad un viso reale quindi ***LBP*** (appliccato con diversi settaggi di raggio e neighbors) può servire nel rilevare queste differenze. A trarne le conclusioni è un ***SVM*** classifier.
- Si può determinare se viene sottomessa una foto poiché quest'ultima avrà una risoluzione inferiore rispetto ad una foto acquisita direttamente dalla fotocamera per l'identificazione
- Controllare il background, poiché se viene mostrata una foto, al muoversi di quest'ultima si muoverà anche tutto il background
- ***Gaze stability***: vengono mostrati all'utente una serie di impulsi sullo schermo in modo random così facendo chi attacca non è in grado di preparare degli attacchi specifici. Ad ogni impulso vengono ripresi i landmark più importanti è tramite questi il sistema determina se è una persona o un'immagine o un video.

Possibili metodi di ***liveness detection*** per 2D face spoofing nei ***video*** sono:

- Detect di un *moirè pattern*, le bande che si presentano quando viene fatta una foto a un display. Le funzioni ***LBP*** e ***DSIFT*** (***Dense Scale Invariant Feature Transform***) vengono utilizzate per rappresentare le caratteristiche dei modelli moiré che differenziano una faccia simulata riprodotta da una faccia reale, individualmente o combinata;



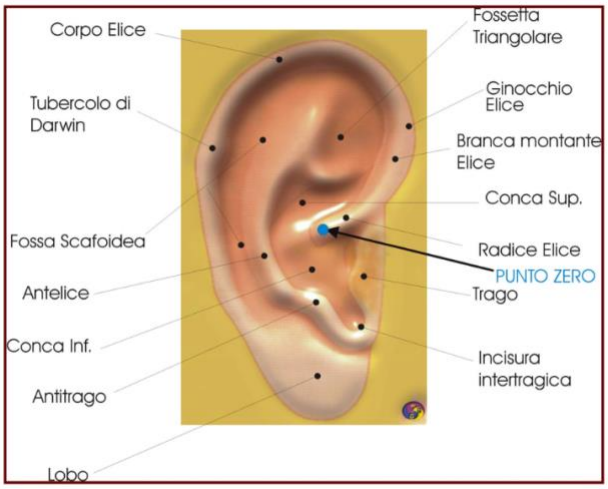
Possibili metodi di ***liveness detection*** per 3D face spoofing è provare a sfruttare le differenze tra autentici ed impostori con proprietà di riflessione della luce sulle maschere. Un altro modo è invece misurare i ***bpm*** della persona rilevando piccoli cambiamenti di colore sotto l'epidermide cutanea tramite ***Fotopletismografia remota*** (***rPPG***). La fotopletismografia remota è una tecnologia

ben studiata che interpreta le variazioni del colore della pelle legato al flusso sanguigno che, se analizzato con un complesso algoritmo matematico, generi letture dei segni vitali.

EAR RECOGNITION

Lezione 9

Esistono molti modi per descrivere un volto, ma pochi modi per descrivere un orecchio.
L'orecchio ha una struttura non-randomica, ma ben definita:



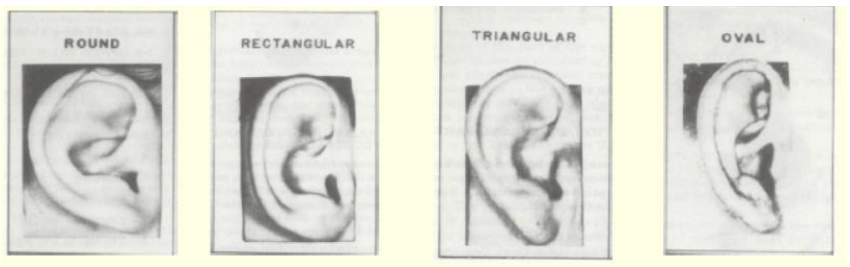
Aspetti positivi:

- L'orecchio è un tratto biometrico passivo e statico.
- Poiché è associato a uno dei sensi umani, di solito viene lasciato scoperto per un migliore udito.
- Se poco visibile è possibile interagire con l'utente.
- L'orecchio è stato confrontato con altri dati biometrici, in particolare il viso. Alcuni vantaggi sono stati evidenziati:
    - meno dettagli, quindi richiedendo una risoluzione inferiore;
    - distribuzione del colore più uniforme.

Un aspetto verificato è la permanenza nel tempo:

- La crescita dell'orecchio è proporzionale dalla nascita ai primi 4 mesi.
- Tra 4 mesi e 8 anni il lobo subisce un maggiore allungamento verticale.
- La dimensione del lobo è costante tra gli 8 ei 70 anni, dopo questo si allunga ulteriormente a causa del rilassamento dei tessuti.

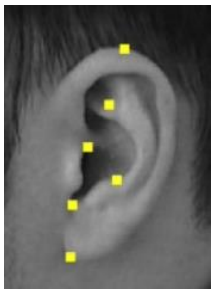
L'orecchio è stato classificato in 4 classi:



L'orecchio ha una dimensione inferiore così come una complessità inferiore rispetto al viso, ma allo stesso tempo ha lo stesso colore della pelle circostante. Questo rende l'estrazione più difficile.

La ***localizzazione*** può essere fatta in differenti modi:

- Un possibile approccio utilizza le reti neurali, che devono essere addestrate e vengono acquisite una grande quantità di immagini.
  Su ogni immagine vengono selezionati i punti di interesse (in giallo):



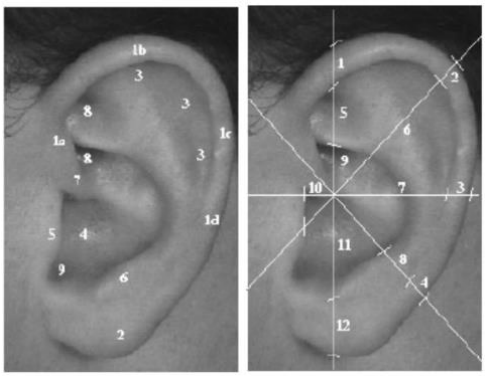
  Poiché l'addestramento per le reti neurali richiede un tempo proporzionale alla dimensione dell'immagine in ingresso, e poiché in questa fase non siamo molto interessati ai dettagli fini, le immagini vengono ridimensionate a una risoluzione inferiore.
  Il contrasto in ogni immagine viene normalizzato per risultati migliori
- Localizzazione dell'oggetto all'interno di un'immagine utilizzando ***AdaBoost***.
- ***Le tecniche 3D*** si basano su una valutazione della profondità e della curvatura delle regioni dell'orecchio. Il *training* è offline e viene effettuato da un modello 3D del profilo del viso, vengono identificati i punti di massima curvatura. Viene creata un'immagine binaria e la regione corrispondente all'orecchio viene estratta manualmente. Le regioni estratte vengono fuse insieme per formare un modello di riferimento. Il *test* viene eseguito online. L'immagine binaria viene calcolata per il nuovo modello, vengono identificati i punti di curvatura massima e minima e vengono ricercate le regioni corrispondenti al modello.

Gli approcci di ***riconoscimento*** possono essere classificati in diversi modi.
Ad esempio, possiamo identificare tre classi di tecniche:

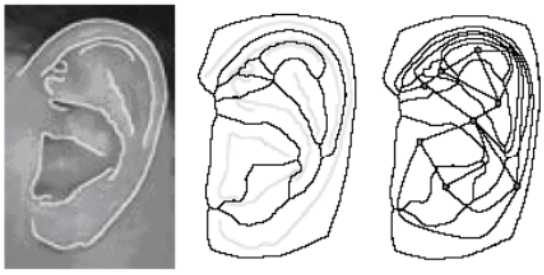
- Approcci geometrici 2D (globali) basati su curve/punti di riferimento
- Modelli 3D
- Termogrammi

APPROCCI GEOMETRICI 2D

- ***Sistema di Iannarelli***: si basa su un insieme di misurazioni fatte sull'orecchio.
  Richiede un allineamento molto accurato e la normalizzazione delle due immagini da confrontare.
  Si individua il crus of helix e lo si fissa come centro del sistema.
  A partire da questo punto si effettuano 12 diverse misurazioni (valori interi).
  Il vettore di feature è costituito dalle informazioni circa il sesso e l'etnia della persona e le 12 misurazioni. Il problema principale con questo metodo è la precisione richiesta in individuando il punto centrale. Se l'identificazione del punto centrale non è corretta, tutte le misure sono sbagliate.



- ***Voroni diagrams***: siccome l'illuminazione rende inefficacie il metodo di Iannarelli, allora *Burge* e *Burger* hanno sottolineato questo problema, proponendo una nuova soluzione, basata sui diagrammi di *Voronoi* e sulla distanza tra grafi. Sebbene il metodo non sia stato sperimentato, la segmentazione dell'orecchio è troppo sensibile a variazioni di illuminazione e posa, rendendo inefficace la rappresentazione di Voronoi.



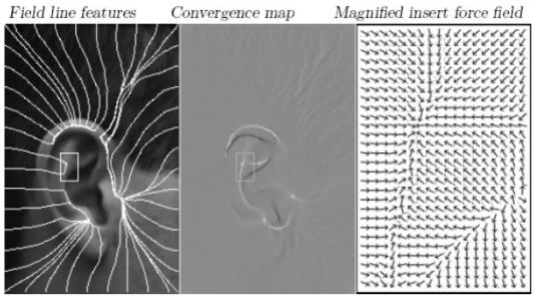
(a) Ear print. (b) Voronoi. (c) N-graph.

- ***Campi di forza***: Ciascun pixel dell'orecchio è visto come una particella carica (0 neutro, 255 carica massima) che esercita un campo di forza (che si assume a simmetria sferica); la forza totale che agisce su un pixel è pari al contributo di tutte le forze dovute agli altri pixel nell'immagine:

$$\mathbf{F}(\mathbf{r}_j) = \sum_{i \neq j} P(\mathbf{r}_i) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

dove r_i rappresenta il vettore posizione del generico pixel
e $P(r_i)$ è il suo valore d'intensità.

Per ciascun pixel si calcola la forza che tutti gli altri pixel esercitano su di esso.
Si fissano una serie di punti su un'ellisse intorno all'orecchio. A partire da ciascun punto si segue l'attrazione del campo di forza. Le linee di campo convergono in punti detti pozzi.
L'orecchio è rappresentato tramite la mappa di convergenza.



MODELLI 3D

Le tecniche 3D si basano su una valutazione della profondità e della curvatura delle regioni dell'orecchio. In alcuni casi il confronto viene effettuato fra regioni corrispondenti (dette patch) di due modelli 3D di orecchio appartenenti alla stessa persona.

TERMOGRAMMA

L'immagine dell'orecchio viene acquisita mediante camera termica

- Vantaggi:
  - L'orecchio è facilmente localizzabile;
  - L'acquisizione è robusta in presenza di occlusione da parte dei capelli;
  - La presenza di colori differenti facilita la segmentazione dell'orecchio.

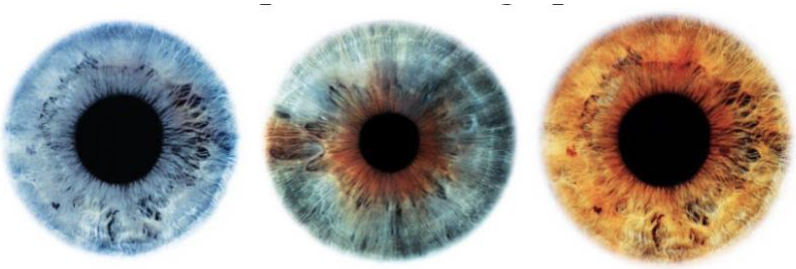
- Svantaggi:
    - Sensibilità al movimento;
    - Limitata risoluzione;
    - Elevati costi.

IRIS RECOGNITION

Lezione 10

L'iride è una membrana muscolare dell'occhio, di colore e forma variabile, con funzione di diaframma. È situata posteriormente alla cornea e davanti al cristallino, perforata dalla pupilla. La tessitura dell'iride si definisce nel corso dei primi due anni di vita ed è caratterizzata da informazioni molto discriminanti, utili ai fini dell'identificazione

Il colore, la tessitura e i pattern dell'iride hanno un elevato grado di individualità paragonabile con quello delle impronte digitali. L'occhio destro e l'occhio sinistro hanno due iridi differenti. Quindi se ci registriamo in un sistema con l'iride destra, l'identificazione dovrà essere fatta sempre con quell'occhio.



Vantaggi dell'iride:

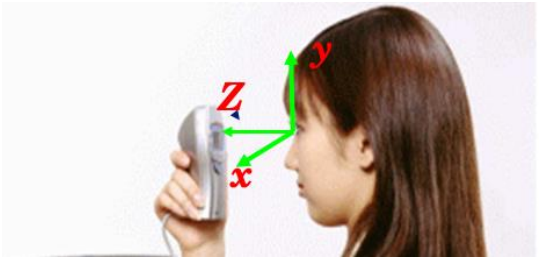
- Nel tempo non varia, è un tratto estremamente distintivo (iride destra diversa da quella di sinistra e anche i gemelli hanno iridi diverse).
- La procedura di acquisizione non è in genere invasiva (tuttavia, alcuni dispositivi richiedono collaborazione da parte del soggetto).
- Permette la creazione di template di piccole dimensioni.

Svantaggi dell'iride:

- La superficie dell'iride è molto limitata, circa 3.64 cm^2.
- Una "*buona*" acquisizione richiede una distanza inferiore a un metro per garantire una risoluzione sufficiente, a seconda del dispositivo di input.

Per quanto riguarda l'acquisizione abbiamo diversi problemi come:

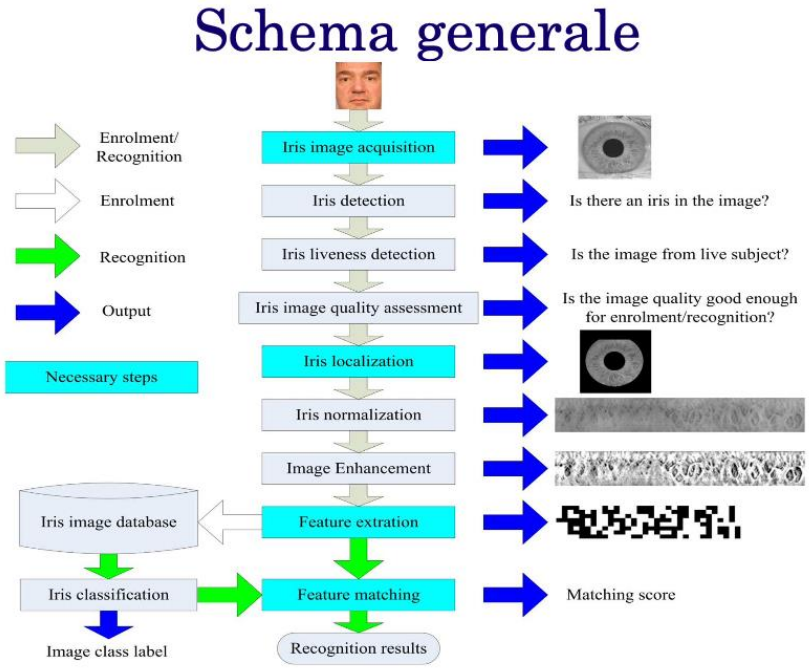
- Piccola dimensione, ovvero circa 11mm;
- Limitata profondità di campo;
- Necessità di essere acquisita allineandosi con un asse;
- Presenza di occhiali
- …



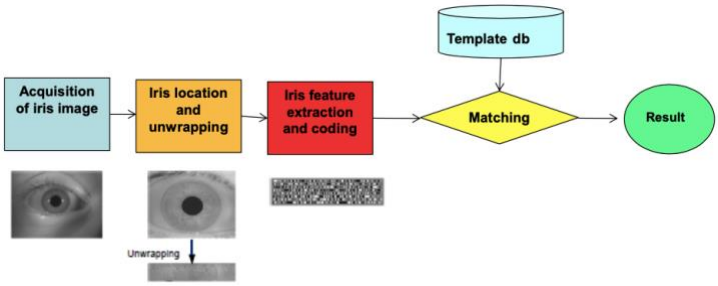
Le modalità di cattura sono due:

- ***Luce visibile***: Gli strati che compongono l'iride sono visibili. L'immagine contiene poche informazioni sulla tessitura. Nella fascia di luce visibile, l'iride rivela una trama molto ricca, casuale, intrecciata (il "reticolo trabecolare")
- ***Luce infrarossi***: La tessitura è più visibile. Più adatta in sistemi biometrici basati sul riconoscimento dell'iride. Nell'illuminazione a infrarossi anche gli occhi marrone scuro mostrano una trama ricca

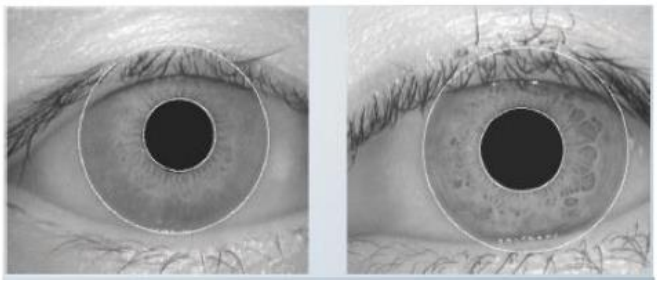
Come avviene il riconoscimento?



DAUGMAN



- ***Posizione dell'iride***: L'approccio utilizza una sorta di rivelatore di bordi circolare per localizzare sia la pupilla che l'iride. L'operatore cerca un percorso circolare lungo il quale viene massimizzata la variazione dei pixel, variando il centro ***r*** e il ***raggio (x0, y0)*** di un profilo circolare candidato. Quando il cerchio candidato ha lo stesso raggio e centro dell'iride, l'operatore dovrebbe fornire un picco.



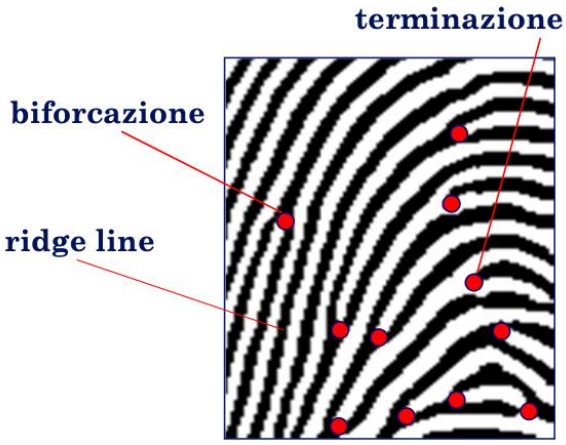
- ***Posizione delle ciglia***: come prima ma rileviamo anche le linee delle palpebre.
- ***Segmentazione dell'iride***: otteniamo una maschera in modo che solo i pixel dell'iride verranno ulteriormente elaborati.
- ***Unwrapping dell'iride***: Determinare il centro giusto per le coordinate polari è di fondamentale importanza ma la pupilla e l'iride non sono perfettamente concentriche e le dimensioni della pupilla possono cambiare a causa dell'illuminazione o di condizioni patologiche (ubriachezza o droghe). La direzione dello sguardo può modificare le posizioni relative della sclera (parte bianca), dell'iride e della pupilla. È necessario elaborare una procedura di normalizzazione: ***Rubber Sheet Model***.
  Il ***Rubber Sheet Model*** è un processo di normalizzazione che tiene infatti conto di fattori quali la dilatazione della pupilla e la deformazione dell'iride. Prendendo un numero fisso di punti su ogni raggio che è contenuto tra il confine della pupilla e il confine dell'iride, è possibile normalizzare la distanza deformata.
  Il modello mappa ogni punto dell'iride in coordinate polari dove il centro delle coordinate polari è il centro della pupilla.
  Durante la trasformazione, le nuove coordinate per ogni punto sono date da una combinazione lineare tra le coordinate del contorno pupillare e quelle del contorno esterno dell'iride.

FINGERPRINTS

Lezione 11

Un'impronta digitale di solito appare come una serie di linee scure che rappresentano la parte alta e appuntita della pelle della cresta di attrito, mentre le valli tra queste creste appaiono come uno spazio bianco e sono la parte bassa e poco profonda della pelle della cresta di attrito.

Le ***minuzie***, o caratteristiche di Galton, sono micro-singolarità determinate dai punti finali o dalle biforcazioni delle linee di cresta, sono delle discontinuità delle ***ridge line***.



La formazione delle impronte digitali è già completa nel *settimo mese* di sviluppo fetale e la configurazione delle creste su ciascun dito è costante durante l'intero ciclo di vita.
La differenza massima tra le impronte digitali si riscontra tra individui appartenenti a razze diverse. Sono seguiti in ordine decrescente della diversità delle impronte da:

- persone della stessa razza ma senza alcuna parentela,
- padre e figlio (che condividono metà dei loro geni),
- fratelli e/o sorelle,
- gemelli.

Esistono due modalità di base per l'acquisizione delle note delle impronte digitali come offline e live-scan:

- ***L'acquisizione off-line*** *avviene in due fasi*.
    - si passano prima i polpastrelli su un tampone di inchiostro e poi si trasferisce l'immagine dell'impronta digitale tramite pressione su carta
    - una successiva digitalizzazione dell'impronta su carta tramite scansione ottica o fotocamera ad alta risoluzione conclude il processo di acquisizione.
- Nel ***live-scan***, l'immagine digitale dell'impronta digitale viene acquisita direttamente tramite il contatto del polpastrello con un apposito sensore.

Le cosiddette ***impronte latenti***, solitamente rilevate sulla scena del crimine, appartengono alla categoria off-line e sono prodotte a causa della natura grassa della pelle che lascia sulla superficie toccata dalle dita una traccia rilevabile successivamente con appositi prodotti chimici reagenti.

Problemi Acquisizione:

- troppo movimento causa distorsione
- pressione variabile sul sensore
- errore nell'estrazione delle feature

Parametri Digitalizzazione:

- Risoluzione
- Area di acquisizione
- Profondità
- Contrasto
- Distorsione geometrica.

Tipi di scanner digitali:

- ***Optical Scanner***: poco costoso, robusto alla variazione di clima e con buona risoluzione, ma è grosso e va pulito molto bene dopo ogni utilizzo;
- ***Capacitive Scanner***: migliore risoluzione dell'impronta e dimensioni ridotte ma dura molto l'acquisizione;
- ***Thermal Scanner***: non può essere ingannato da impronte artificiali perché riconosce pulsazione, temperatura, pori e cambiamento di colore della pelle tramite pressione, anche se l'immagine scompare rapidamente, è infatti possibile riprodurre Fake Fingerprints tramite gelatina, silicone e lattice.

Il confronto tra impronte digitali è abbastanza difficile per i seguenti motivi:

- Scarsa sovrapposizione: non sempre si assume la stessa posizione assunta quando si è registrata per la prima volta l'impronta.
- Spesso si hanno condizioni della pelle diverse (più o meno sudate/ grasse)

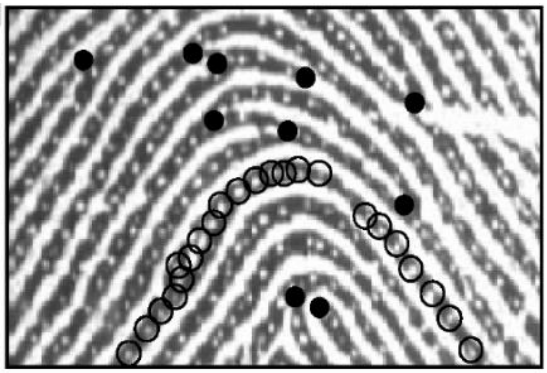
- Elevata distorsione


scarsa sovrapposizione


condizioni della pelle molto diverse


elevata distorsione non lineare

Con il matching per ***minuzie*** possiamo trovare anche dettagli intra-creste, che sono essenzialmente i pori della pelle per la sudorazione, la cui posizione e forma sono considerate estremamente distintive.
Purtroppo, l'estrazione dei pori è possibile solo a partire dalle immagini dell'impronta digitale acquisite ad altissima risoluzione, dell'ordine dei 1000 dpi, e in condizioni ideali, quindi questa particolare rappresentazione non è praticabile per la maggior parte dei contesti applicativi.



TECNICHE DI CONFRONTO

- ***matching basato sulla correlazione***: le due immagini vengono sovrapposte e viene iterato il calcolo della correlazione tra pixel corrispondenti per diversi allineamenti, che si ottengono tramite roto-traslazioni fino a determinare il grado di similarità tra i campioni; sensibile alle trasformazioni rigide e non lineari; elevata complessità computazionale
- ***matching basato sulle caratteristiche delle creste***: l'estrazione di minuzie nelle immagini delle impronte digitali di bassa qualità è problematica, quindi questi metodi utilizzano altre caratteristiche come l'orientamento delle creste e la frequenza locale, la forma delle creste e la trama, che sono più affidabili e più facili da estrarre, ma anche meno distintivo; basso potere discriminante
- ***matching basato su minuzie***: le minuzie vengono prima estratte dalle due impronte e memorizzate come due insiemi di punti in uno spazio bidimensionale (eventualmente annotato con l'angolo tra la tangente e il piano orizzontale), quindi i metodi ricercano l'allineamento tra i due set che massimizza il numero di coppie corrispondenti di minuzie, e in base a questa misura la somiglianza tra le impronte digitali (point pattern matching)

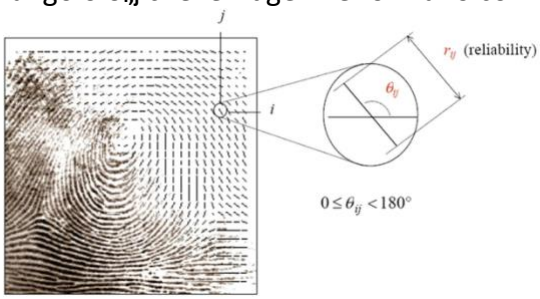
FEATURE EXTRACTION

- ***Segmentation***: il termine segmentazione è solitamente utilizzato per indicare la separazione dell'area dell'impronta (***foreground***) dallo sfondo (***background***). Il foreground si caratterizza per la presenza di un pattern striato e orientato, mentre il background presenta un pattern ***isotropico***, ovvero presenta sempre le stesse caratteristiche (infatti lo sfondo è sempre bianco). Anisotropia invece è la proprietà di essere direzionale

dipendente, al contrario dell'isotropia, che implica proprietà identiche in tutte le direzioni. Come la misuro:

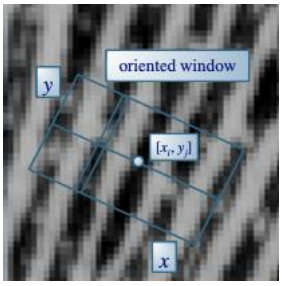
- Presenza di un picco in un istogramma locale di orientazioni
- Varianza dei livelli di grigio
- Combinazione di più caratteristiche, andandole a calcolare sui pixel

- ***Directional Map***: L'orientazione locale delle ridge line in posizione [i,j] è definita come l'angolo $\theta_{i,j}$ che le ridge line formano con l'asse orizzontale.



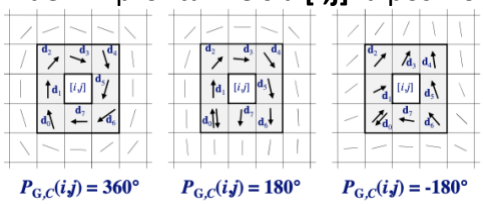
Invece che calcolare l'orientazione per ogni singolo punto, molto spesso si preferisce velocizzare la procedura, calcolandola in corrispondenza di posizioni discrete.
RIVEDI

- ***Frequency Map***: Si studia per ogni punto presente in una impronta digitale il numero di creste per unità di lunghezza lungo un ipotetico segmento centrato in ***[x, y]*** e ortogonale all'orientamento della cresta locale.



Un possibile approccio consiste nel contare il numero medio di pixel tra picchi consecutivi di livelli di grigio lungo la direzione ortogonale all'orientamento locale della linea di cresta.
In questo caso avremo 5 picchi, uno ogni volta che si incontra una cresta.

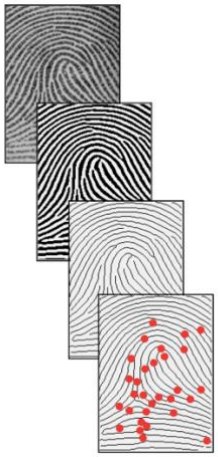
- ***Singolarità***: la maggior parte degli approcci si basa sull'immagine direzionale dell'impronta. Un metodo pratico ed elegante si basa sul calcolo dell'indice di ***Poincaré***. Sia ***G*** un campo vettoriale e sia ***C*** una curva immersa in ***G***; l'indice di Poincaré $P_{G,C}$ è definito come la rotazione totale dei vettori di ***G*** lungo ***C***. Sia **G** il campo associato all'immagine delle orientazioni dell'impronta **D** e sia **[i,j]** la posizione dell'elemento θ_{ij} nell'immagine:



$$P_{G,C}(i,j) = \sum_{k=0..7} angle\left(\mathbf{d}_k, \mathbf{d}_{(k+1) \bmod 8}\right)$$

L'indice $P_{G,C}(i,j)$ si calcola ***sommando algebricamente le differenze di orientazione tra elementi adiacenti*** di C.

- ***Minuzie***: in generale, l'estrazione delle minuzie comporta:



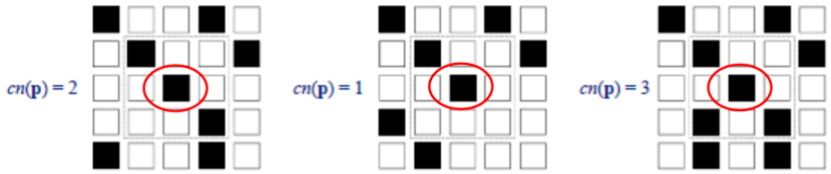
Binarizzazione: conversione di un'immagine a livelli di grigio in un'immagine binaria;
Thinning: l'immagine binaria è sottoposta a un passo di assottigliamento che riduce lo spessore delle ridge line a 1 pixel;
Localizzazione: si fa ricorso a una scansione dell'immagine per localizzare i pixel corrispondenti alle minuzie

La ***localizzazione*** delle minuzie si basa sull'analisi del crossing number:

$$cn(\mathbf{p}) = \frac{1}{2}\sum_{i=1..8}\left|val(\mathbf{p}_{i \bmod 8}) - val(\mathbf{p}_{i-1})\right|$$

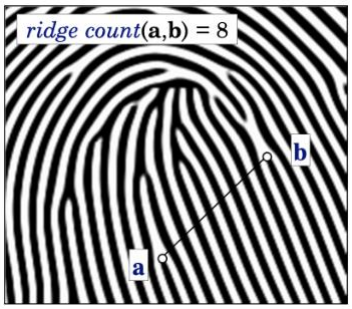
dove p_0, p_1, ..., p_7 sono i pixel appartenenti a una sequenza ordinata di pixel che costituiscono l'intorno del pixel ***p*** e $val(p) \in \{0,1\}$ è il valore del pixel ***p***. Si nota che un pixel ***p*** con ***val(p)=1***:

- è un **punto interno** a una ridge line se **cn(p)=2**;
- corrisponde a una **terminazione** se **cn(p)=1**;
- corrisponde a una **biforcazione** se **cn(p)=3**;
- appartiene a una **minuzia più complessa** se **cn(p)> 3**.



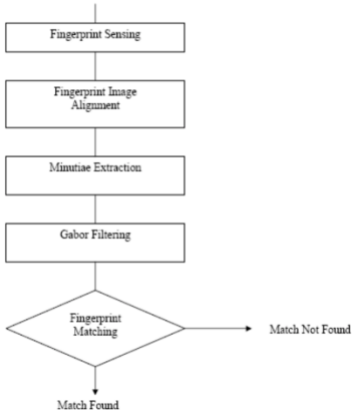
Punto interno (ifattti se vediamo continua in basso a destra)
Terminazione perché dopo non ci sono più punti neri
Biforcazione poiché sia a destra che a sinistra continua

- Stima del numero di ridge line: tra le caratteristiche delle impronte usate dagli esperti per il riconoscimento c'è anche il ***numero di ridge*** (***ridge count***). Il ridge count è una misura astratta della distanza tra due punti qualsiasi di un'impronta. Presi due punti ***a*** e ***b*** di un'impronta, il ridge count corrisponde al numero di ridge line ***intersecate dal segmento ab***.



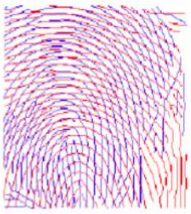
Sebbene il ridge count sia definito per due punti qualsiasi, solitamente ***a*** e ***b*** sono scelti in corrispondenza di punti fissi dell'impronta digitale (es. posizione di singolarità o minuzie). Ad esempio, in ambito forense spesso si conteggiano le ridge tra core e delta

MATCHING



Un possibile approccio ibrido al confronto delle impronte digitali combina la rappresentazione delle impronte digitali basata su ***minuzie*** con una rappresentazione basata sul ***filtro di Gabor*** che utilizza informazioni sulla trama locale.

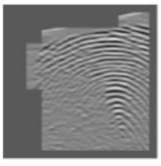
Same finger

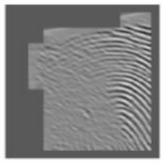


Different fingers

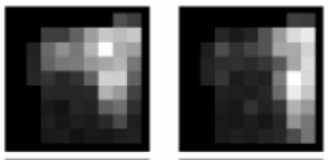
La ***fase di allineamento*** inizia con l'estrazione delle minuzie sia dall'input che dal template da abbinare. Le due serie di minuzie vengono confrontate attraverso un algoritmo di point matching che seleziona preliminarmente una coppia di minuzie di riferimento (una da ciascuna immagine), e quindi determina il numero di coppie di minuzie utilizzando l'insieme di punti rimanente. La coppia di riferimento che produce il numero massimo di coppie corrispondenti determina il miglior allineamento.

Le aree dell'immagine di sfondo dell'input del polpastrello vengono mascherate come non necessarie.





Per eseguire l'estrazione delle caratteristiche dalle celle risultanti dalla tassellatura viene utilizzato un gruppo di ***8 filtri Gabor***, tutti con la stessa frequenza, ma con orientamento variabile. Tale filtraggio produce 8 immagini ordinate per ogni cella.



Feature extraction: I valori caratteristici di tutte le celle vengono quindi concatenati in un ***vettore caratteristico***. I valori caratteristici relativi alle regioni mascherate nell'immagine in ingresso non vengono utilizzati per la successiva fase di confronto e sono contrassegnati come valori mancanti nel vettore caratteristico.

Infine il ***confronto*** dell'immagine in ingresso con il template memorizzato viene effettuato calcolando la somma dei quadrati delle differenze tra i corrispondenti vettori caratteristici, dopo aver scartato i valori mancanti. Se il punteggio di somiglianza è inferiore a una soglia, è possibile affermare che l'immagine di input ha un modello corrispondente in memoria e il riconoscimento ha esito positivo.

Non è poi così difficile riprodurre impronte false, soprattutto di un utente cooperativo. I materiali più popolari sono gelatina, silicone, lattice. Ogni materiale produce impronte di diversa qualità e con caratteristiche diverse.

LIVENESS DETECTION

Una misura di sicurezza particolarmente interessante per contrastare eventuali tentativi di imbroglio basati sulle impronte digitali dei sistemi, è quella di determinare se la sorgente del segnale in ingresso (il dito) sia un vero tratto biometrico vivente piuttosto che il simulacro di un'attività fraudolenta (un calzata del guanto recante l'impronta digitale di un utilizzatore autorizzato).
La premessa logica di un test di rilevamento della vivacità è che se il dito è vivo, la sua impronta è in realtà della persona a cui appartiene.
Uno degli approcci più comuni al test di vitalità consiste nell'utilizzare uno o più parametri vitali comuni a tutta la popolazione di riferimento, come ad esempio polso e temperatura.
Gli scanner ottici di tipo live-scan con tecnologia FTIR (Frustrated Total Internal Reflection) utilizzano un meccanismo di acquisizione differenziale per le creste e i solchi delle impronte digitali, risultando in questo modo intrinsecamente più resistenti agli attacchi di un simulacro con impronta bidimensionale del dito.
La scansione ad alta risoluzione dell'impronta digitale rivela dettagli caratteristici della struttura dei pori che sono molto difficili da imitare in un dito artificiale.
Il colore caratteristico della pelle del dito cambia a causa della pressione quando viene premuto sulla superficie di scansione e questo effetto può essere rilevato per identificare l'autenticità del dito. Anche il flusso sanguigno e la sua pulsazione possono essere rilevati con un'attenta misurazione della luce riflessa o trasmessa attraverso il dito.
La differenza di potenziale tra due punti specifici della muscolatura del dito può essere utilizzata per distinguerlo da un dito morto.
La misurazione dell'impedenza complessa del dito può essere utile per verificare la vitalità del dito.
Infine, un dito suda e questo può determinare la vitalità del dito.

MULTIBIOMETRICS SYSTEMS

Lezione 12

Sono sistemi che integrano più sorgenti di dati biometrici al fine di migliorare le prestazioni di riconoscimento.
Multibiometrics vuol dire diverse cose:

- ***Multipli algoritmi***: ad esempio 2 algoritmi per il riconoscimento delle impronte come Filterbank e Minutie.
- ***Multipli sensori***.
- ***Multipli traits***: ad esempio firma e impronte digitali.
- ***Multiple istance***: impronta digitale dell'indice e del medio.

La combinazione può essere fatta in diversi modi:

Prima del matching:

- ***Fusione a livello di sensore***:
  - I dati acquisiti da sensori diversi possono essere elaborati e integrati per generare nuovi dati dai quali si etraggono poi le feature.
    - Per esempio, nel caso di riconoscimento del volto, si possono fondere le informazioni 2D (tessitura) e quelle 3D (range image), ricercare sensori diversi, per generare un modello 3D completo.
- ***Fusione a livello di feature***:
  - Le feature estratte con tecniche diverse possono essere fuse per creare un nuovo vettore di feature rappresentativo dell'individuo.
    - Le caratteristiche geometriche della mano, per esempio, possono essere abbinate alle feature estratte dal volto usando il metodo Eigenface, ottenendo così un vettore unico. Si possono poi eventualmente applicare tecniche di selezione delle feature per ridurre la dimensionalità del vettore, mantenendo solo le informazioni più significative.

Dopo il matching:

- ***Fusione a livello di score***:
  - Algoritmi di matching diversi restituiscono un insieme di score che vengono poi fusi per generare un unico score finale.
    - Ad esempio gli score ottenuti dal matching dell'impronta e del volto possono essere combinati usando la regola della somma per ottenere un singolo score.
- ***Fusione a livello di rank***:
  - Questo tipo di fusione è utile in sistemi di identificazione; in questo caso diversi classificatori forniscono un ranking (ordinamento) delle classi (rank elevato indica un buon match).
    - I rank dei diversi classificatori sono unificati per ottenere una "classifica finale" utile per la decisione sull'identità della persona (es. Borda count).
- ***Fusione a livello di decisione***:

- Ogni classificatore restituisce in output la propria decisione (accept/reject in caso di verifica o l'identità in caso di identificazione). La decisione finale è presa combinando le singole decisioni a seconda di una regola (es. maggioranza dei voti).

Per quanto riguarda la ***fusione a livello di score***, ogni classificatore fornisce in output la propria decisione che consiste della classe cui ha assegnato il pattern e del livello di confidenza. Le decisioni possono essere combinate in diversi modi:

- ***Majority vote rule***: ogni classificatore vota per una classe, il pattern viene assegnato alla classe maggiormente votata.
- ***Borda count***: ogni classificatore produce una classifica o ranking delle classi (dalla prima all'ultima) a seconda della probabilità che il pattern appartenga a ciascuna di esse. I ranking sono poi convertiti in punteggi che sono tra loro sommati; la classe con il più elevato punteggio finale è quella scelta dal multi- classificatore.

Data la qualità differente che possiamo avere dai diversi sottosistemi è opportuno stabilire delle misure di affidabilità prima di trarre la nostra conclusione.
Una possibile soluzione è quella di applicare dei margini di confidenza basati sulle stime del FAR e FRR: $M(\Delta) = |FAR(\Delta) - FRR(\Delta)|$

DEFINIZIONI

- **Probe**: modello biometrico generato ogni volta che un interessato interagisce con il sistema biometrico.
- **Template**: un insieme di caratteristiche biometriche memorizzate, confrontabili direttamente con altri modelli biometrici.
- **Gallery**: è l'insieme di templates appartenenti a vari soggetti nel databse.
- **Ground Truth**: quando si etichettano i campioni, per gli esperimenti, con l'identità corretta.