

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE
INFORMATICHE

**IL RICONOSCIMENTO DEL VOLTO
IN PRESENZA DI ALTERAZIONI DIGITALI
DELLE IMMAGINI**

Relazione finale in
BASI DI DATI

Relatore
Dott.ssa Annalisa Franco

Presentata da
Andrea Zanuccoli

Sessione III

Anno Accademico 2013-2014

Indice

Introduzione	1
Capitolo 1: Il Morphing	5
1.1 Definizione matematica di morphing	6
1.2 Il Warping	8
1.2.1 Trasformazioni parametriche.....	9
1.2.2 Trasformazioni non parametriche.....	14
1.3 Face Morphing : I sistemi automatici allo stato dell'arte	22
Capitolo 2: Riconoscimento del volto	25
2.1 Face Detection	27
2.1.1 Skin Segmentation	28
2.1.2 Il metodo Viola-Jones	29
2.2 Estrazione delle feature	32
2.2.1 Dimensionality Reduction.....	33
2.2.2 Filtri di Gabor	34
2.3 Riconoscimento delle feature	36
2.3.1 Eigenfaces.....	37
2.3.2 Support Vector Machines.....	39
2.3.3 Reti neurali.....	40
2.3.4 Hidden Markov Model.....	46
Capitolo 3: Riconoscimento del volto in presenza di alterazioni digitali	
delle immagini	49
3.1 Distorsioni geometriche e digital beautification	49
3.2 Morphing	52
3.3 Morphing Software	52
3.3.1 Morphthing	53

3.3.2 Fotomorph	55
3.3.3 Sqirlz Morph.....	57
3.3.4 Face Morpher Lite	57
3.3.5 Face On Face.....	59
3.3.6 Confronto tra i software di morphing	60
3.4 Test con software commerciali.....	63
3.4.1 VeriLook 5.3/MegaMatcher 4.4 Algorithm Demo.....	65
3.4.2 Luxand SDK Face Matcher.....	67
3.5 Risultati.....	68
Conclusioni.....	79
Biografia e Sitografia	81
Elenco delle figure	83

Introduzione

L'Organizzazione Internazionale dell'Aviazione Civile (ICAO), ovvero l'agenzia delle Nazioni Unite incaricata della gestione delle normative riguardanti la sicurezza, la regolarità e l'efficacia della navigazione aerea internazionale, ha annunciato che dal 24 Novembre 2015 i passaporti diversi da quelli a lettura ottica (MRTD) non potranno più essere utilizzati come documenti di viaggio validi. [1]

Ciò significa che i vecchi passaporti non rispetteranno più le normative definite dall'ICAO e sarà necessario disporre di un documento leggibile da una macchina per poter viaggiare.

I passaporti a lettura ottica (elettronico) possono essere scansionati da una macchina in grado di leggere le informazioni contenute nel microchip che si trova all'interno del documento, nel quale sono memorizzati i dati in forma digitale, e concede la possibilità agli agenti doganali o altre forze della polizia di esaminare i documenti rapidamente, senza dover introdurre manualmente le informazioni contenute in un computer, al fine di poter controllare in un database se il passaporto è stato rubato o se il possessore non è ricercato.

Nel microchip sono memorizzati i consueti dati anagrafici e, nel caso di passaporti biometrici, l'immagine del volto – che è stato scelto come caratteristica biometrica primaria per la verifica di identità – e opzionalmente le immagini di impronte digitali e iride.

Per facilitare il riconoscimento del volto, la fototessera inclusa nel passaporto deve soddisfare i requisiti geometrici e fotometrici standard forniti dall'ICAO e può essere fornita in due modi (in base al paese in cui viene rilasciato il passaporto): acquisita da una fotocamera digitale ad alta risoluzione dall'ufficio che rilascia il documento, o sotto forma di fototessera fornita dal cittadino stesso.

Quando la foto viene fornita dal cittadino l'ufficiale che rilascia il passaporto controlla che questa soddisfi i requisiti di qualità dettati dallo standard ISO (*International Standard Organization*) [2] ma non effettua un ulteriore controllo che risulterà fondamentale nell'identificazione del volto: che l'immagine non sia stata alterata.

L'inserimento nel documento di una foto del volto alterata potrebbe creare numerosi problemi, in particolare nel caso di riconoscimento automatico, non supervisionato da un esperto umano. Questa ipotesi è molto rilevante in considerazione della presenza di *e-Passport Gates* (Figura 1.1), noti anche come sistemi ABC (*Automated Border Control*) installati in molti aeroporti dei paesi più sviluppati che forniscono un'alternativa al controllo dei funzionari d'immigrazione.



Figura 1.1: Esempio di e-Passport Gates

Il sistema d'identificazione è fornito di un lettore ottico su cui viene posizionato il passaporto per la lettura dei dati; in seguito una fotocamera scatta una foto al possessore del passaporto, ed insieme alla verifica dei dati il sistema effettuerà un riconoscimento del volto confrontando l'immagine rilevata dal documento con la foto scattata.

Tali sistemi di riconoscimento potrebbero essere in qualche modo "ingannati" facendo uso di immagini alterate. In particolare in questo lavoro di tesi si valuta la possibilità di attaccare i sistemi ABC tramite immagine del volto risultanti da un processo di *morphing*.

Un'immagine *morphed* di un volto è il risultato della combinazione dei volti di due o più soggetti; qualora in tale immagine fossero riconosciuti (dai software di

riconoscimento del volto) tutti i soggetti usati nella combinazione dei volti, un passaporto in cui è inserita la foto potrebbe essere utilizzato da più di una persona.

Partendo da un database di volti di persone, impiegando cinque software freeware per *morphing*, e due software per il riconoscimento del volto che simuleranno il controllo svolto da un *e-Passport Gates*, l'obiettivo di questo lavoro di tesi è provare l'efficacia dell'utilizzo di una foto *morphed* per un attacco a un sistema ABC.

La tesi è organizzata come segue.

Nel primo capitolo verrà illustrato il processo di sviluppo del *morphing*, descrivendo i principali metodi, e prestando particolare attenzione alla deformazione delle immagini richiesta da questo effetto digitale.

Nel secondo capitolo saranno delineati i concetti fondamentali del riconoscimento del volto, illustrando la localizzazione del viso di una persona, l'estrazione delle caratteristiche principali, e i gli algoritmi utilizzati con maggior frequenza.

Nel terzo capitolo vengono presentati i software utilizzati nella ricerca e i risultati del riconoscimento dei soggetti usati nei *morphing* corrispondenti.

Capitolo 1

Il Morphing

Il termine *morphing* deriva dalla parola inglese “*metamorphosis*” (in italiano metamorfosi, che significa cambiamento di forma) e viene usato per indicare quelle tecniche che permettono di creare una sequenza di immagini che se inserite in un’animazione mostrano la graduale trasformazione da un soggetto iniziale detto *source* ad uno finale nominato *target*.

Il *morphing* è uno dei primi effetti digitali sviluppati, è nato nella seconda metà degli anni sessanta per scopi industriali e militari, ma il suo impiego massiccio è partito negli anni novanta nel cinema in film come *Terminator 2* e *Indiana Jones e l’ultima crociata*. Prima di allora nell’industria cinematografica le tecnologie informatiche non erano sufficientemente evolute da permettere produzioni di qualità in tempi ragionevoli e con costi sostenibili, e la trasformazione di un’immagine in un’altra era resa possibile dalla dissolvenza incrociata (Figura 2.1), in cui l’immagine di partenza si dissolve gradualmente in nero, cioè scompare, e nello stesso momento si effettua l’assolvenza, sempre in modo graduale, dell’immagine finale, che quindi appare. Più precisamente in questo metodo il colore di ogni pixel della prima immagine viene *interpolato* con il colore del pixel corrispondente nella seconda immagine, cioè viene sostituito in modo progressivo con i valori intermedi di colore della coppia di pixel corrispondenti nelle due immagini.



Figura 2.1: Dissolvenza incrociata fra due immagini

L'utilizzo di questa tecnica in una trasformazione crea un problema evidente quando i contorni dei soggetti *source* e *target* sono molto distinti, poiché non permette la deformazione delle immagini, rendendo impossibile l'allineamento dei tratti del viso dei due soggetti (occhi, viso, bocca, naso, fronte) per rendere l'animazione molto più realistica.

La dissolvenza incrociata è presente anche nella tecnica del *morphing* ma la fase di dissolvenza è preceduta da un'altra chiamata *warping*.

Il *warping* è un processo di manipolazione digitale di un'immagine in cui viene modificata la posizione dei punti di un'immagine senza cambiarne i colori, realizzando così una deformazione.

Nel *morphing* la deformazione dei punti di un'immagine è necessaria per far sì che i tratti della figura *source* assomiglino il più possibile a quelli della figura *target* in modo tale che una volta completata la deformazione i contorni delle due figure siano perfettamente sovrapposti in modo da ottenere nella dissolvenza incrociata un'animazione fluida e graduale. [3]

1.1 Definizione matematica di morphing

Un oggetto grafico consiste in una qualsiasi entità processata in un sistema di computer grafica. Inspirato da un oggetto del mondo fisico, un oggetto grafico può avere forma e attributi, ad esempio i materiali di cui è costituito o il suo colore.

La definizione matematica di un oggetto grafico è la seguente:

Un oggetto grafico O di uno spazio euclideo R_n , consiste in un sottoinsieme $U \subset R_n$ e una funzione $f: U \rightarrow R_p$.

L'insieme U definisce la forma o le informazioni geometriche dell'oggetto, mentre la funzione f definisce gli attributi dell'oggetto grafico, per questa ragione viene chiamata funzione attributo dell'oggetto.

Si usa l'annotazione $O(U, f)$ quando ci si riferisce all'oggetto grafico considerando sia la sua forma che i suoi attributi.

Quando si vuole effettuare una trasformazione di un oggetto grafico, nel caso del *morphing* usando il *warping* e la dissolvenza incrociata, si possono modificare due tipi di famiglie di parametri che costituiscono l'oggetto: *domain* (dominio) e *range* (codominio). [4]

In matematica una funzione è il dato di tre oggetti: un *dominio* X e un *codominio* Y , che sono gli insiemi su cui è definita la funzione, e una *legge* $X \rightarrow f(X)$ che associa ad ogni elemento del dominio X uno e un solo elemento del codominio Y che viene indicato $f(X)$.

Il *warping* è una tecnica che mira a deformare i contorni, i tratti, e quindi le forme di un oggetto grafico; ciò avviene modificando le coordinate dei punti (pixel) dell'immagine realizzando quella che viene definita una *mappatura*, nella quale le relazioni spaziali dei punti dell'immagine non vengono mantenute.

L'alterazione del sottoinsieme U che definisce appunto la forma dell'oggetto prende il nome di *domain transformation* (Figura 2.2):

$$g(x) = f(T(X))$$

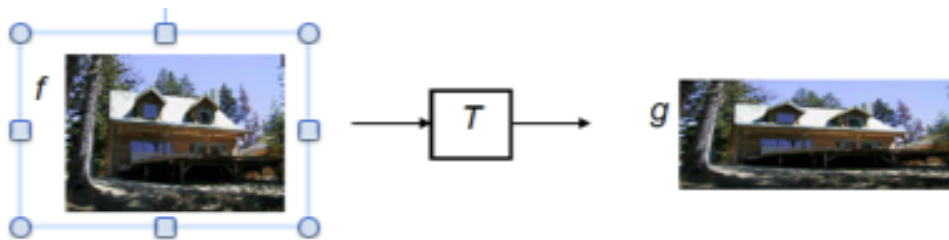


Figura 2.2: Domain transformation di un'immagine

Al contrario la dissolvenza incrociata modifica i colori di un oggetto grafico, che fanno parte degli attributi, lasciando invariate le relazioni spaziali fra i punti perciò in questo caso viene alterata la funzione f che definisce gli attributi di un oggetto grafico; questa alterazione prende il nome di *range transformation* (Figura 2.3):

$$g(x) = T(f(x))$$

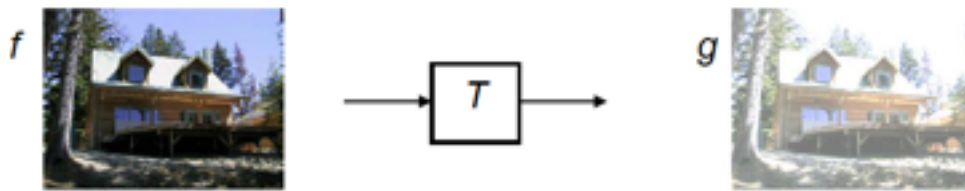


Figura 2.3: Range transformation di un'immagine

1.2 Il Warping

Con il termine *warping* si definisce ogni tecnica che produce una trasformazione geometrica dei pixel di un'immagine.

Tecniche di *warping* sono molto utilizzate in numerose applicazioni di analisi di immagini, come per la rimozione di distorsioni ottiche introdotte da una macchina fotografica, o l'allineamento nel caso del *morphing*.

In questi casi si effettua una mappatura dei punti dell'immagine sorgente nelle nuove posizioni all'interno dell'immagine finale.

Il *mapping*, nello specifico, è la fase che consiste nel determinare la funzione che stabilisce la corrispondenza tra la posizione dei pixel nell'immagine sorgente e la posizione alterata nell'immagine finale.

Esistono due tipi di *warping* basati su trasformazioni di tipo diverso: *trasformazioni parametriche* (globali) e *trasformazioni non parametriche* (locali). [4]

1.2.1 Trasformazioni parametriche

Le trasformazioni parametriche definiscono una funzione unica di forma stabilita che si applica per ogni punto dell'immagine e ne modifica la posizione. Fanno parte di questa categoria tutte le trasformazioni geometriche comuni in due dimensioni.

La trasformazione parametrica più generica è la trasformazione affine.

Le trasformazioni affini sono una qualunque composizione di trasformazioni lineari in cui le rette si trasformano in rette, rette parallele corrispondono a rette parallele e rette incidenti corrispondono a rette incidenti.



Figura 2.4: Esempi di trasformazione affine: traslazione, rotazione, riscalatura, affine

Nelle trasformazioni affini non si conservano la forma delle figure e l'ampiezza degli angoli.

Fanno parte di questa categoria la traslazione, la rotazione, la riscalatura e le riflessioni (Figura 2.4).

Il *mapping*, cioè la trasformazione delle coordinate di un punto da (x,y) dell'immagine sorgente a (u,v) dell'immagine di destinazione è costituita da due equazioni di sei parametri:

$$u = a_{10}x + a_{01}y + a_{00}$$

$$v = b_{10}x + b_{01}y + b_{00}$$

Un'altra famiglia di trasformazioni parametriche sono le trasformazioni proiettive, in cui i punti dell'immagine sorgente vengono proiettati attraverso delle rette su un piano dove si genera l'immagine di destinazione.

Fanno parte di queste trasformazioni le proiezioni prospettiche, nelle quali esiste una distanza finita tra il centro di proiezione e il piano, e le proiezioni parallele, dove la distanza tra il centro di proiezione e il piano è infinita (Figura 2.5).

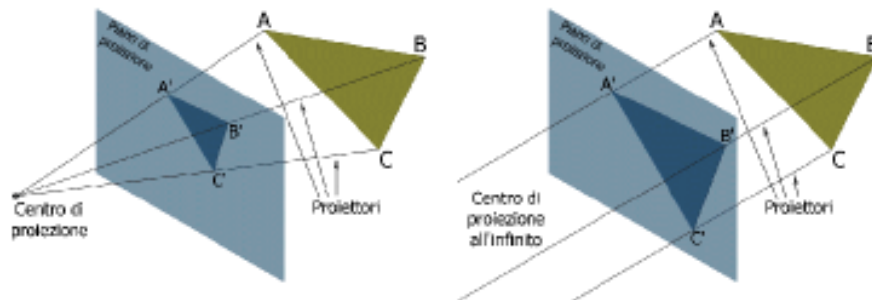


Figura 2.5: Esempi di proiezione prospettica a sinistra e proiezione parallela a destra

Una trasformazione prospettica dei punti di una figura è costituita da due equazioni di otto parametri:

$$c_{10}xu + c_{01}yu + u = a_{10}x + a_{01}y + a_{00}$$

$$c_{10}xv + c_{01}yu + v = b_{10}x + b_{01}y + b_{00}$$

Per realizzare questo tipo di trasformazioni è possibile disporre delle seguenti tre tecniche di interpolazione: affine, bilineare e proiettiva.

Interpolazione Affine

L'interpolazione affine utilizza i triangoli per determinare le nuove coordinate di un punto.

Consideriamo due triangoli ABC e DEF di un spazio euclideo, una trasformazione affine è una mappatura lineare dei punti di un triangolo a quelli dell'altro triangolo (Figura 2.6).

Assumiamo che i vertici del triangolo DEF siano la trasformazione dei vertici di ABC , definiti come segue $T(A) = D$, $T(B) = E$ e $T(C) = F$, e che ogni pixel del triangolo ABC sia la seguente combinazione lineare:

$$p = \lambda_1 A + \lambda_2 B + \lambda_3 C, \text{ dove } \lambda_i \geq 0, \text{ e } \lambda_1 + \lambda_2 + \lambda_3 = 1.$$

Allora la trasformazione di $q = T(p)$ è ottenuta così:

$$q = T(p) = T(\lambda_1 A + \lambda_2 B + \lambda_3 C) = \lambda_1 T(A) + \lambda_2 T(B) + \lambda_3 T(C) = \lambda_1 D + \lambda_2 E + \lambda_3 F.$$

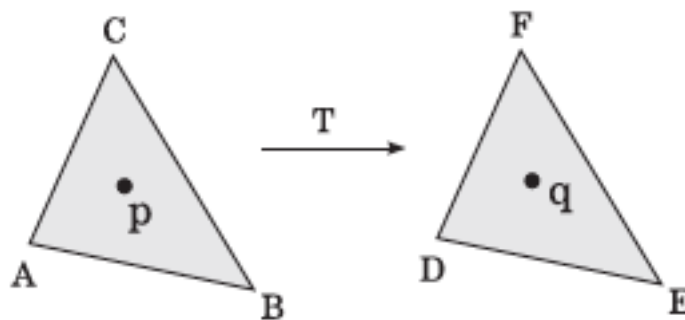


Figura 2.6: Trasformazione con interpolazione affine

Interpolazione Bilineare

A differenza dell'interpolazione affine quella bilineare si serve dei quadrilateri.

Considerando due quadrilateri $ABCD$ e $EFGH$, per ogni punto p di $ABCD$ si assume che la posizione di p è una combinazione lineare dei vettori A , B , C e D (Figura 2.7).

La trasformazione bilineare è data dalle seguenti equazioni:

$$p = (1-u)(1-v)A + u(1-v)B + uvC + (1-u)vD \text{ con } 0 \leq u, v \leq 1$$

e quindi q sarà data da:

$$q = (1-u)(1-v)E + u(1-v)F + uvG + (1-u)vH.$$

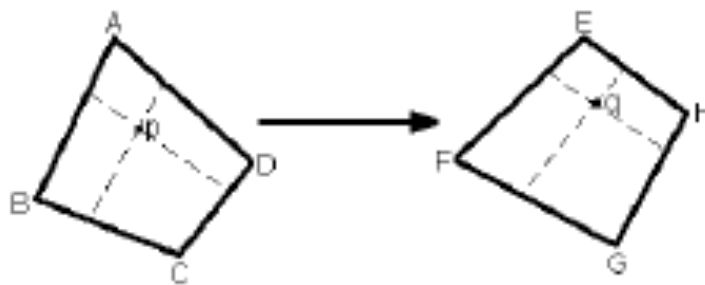


Figura 2.7: Trasformazione con interpolazione bilineare

Interpolazione Proiettiva

Dati due sistemi di riferimento bidimensionali un punto P di coordinate x,y diviene il punto $P'(x',y')$ attraverso le seguenti equazioni:

$$X = \frac{a_1x + a_2y + a_3}{c_1x + c_2y + 1}$$

$$Y = \frac{b_1x + b_2y + b_3}{c_1x + c_2y + 1}$$

Confronto

Per mostrare le differenze fra le diverse tecniche, vediamo come possono modificare le forme di una scacchiera (Figura 2.8).

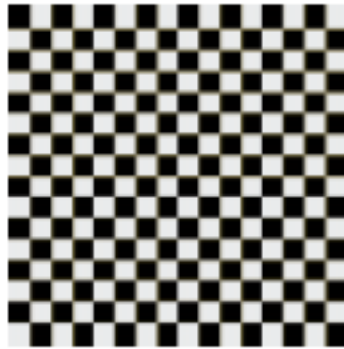


Figura 2.8: Immagine di una scacchiera

Utilizzando la trasformazione affine la scacchiera viene suddivisa in due triangoli, ciò è dovuto alla discontinuità della derivata prima che influenza la trasformazione delle coordinate dei punti.

Con la trasformazione bilineare i segmenti della scacchiera si trasformano in curve a causa del sistema di equazioni di secondo grado.

La trasformazione proiettiva molto spesso è confusa con quella bilineare, ma osservando le immagini si nota che al contrario di quella bilineare nella proiettiva le rette si trasformano in rette (Figura 2.9).

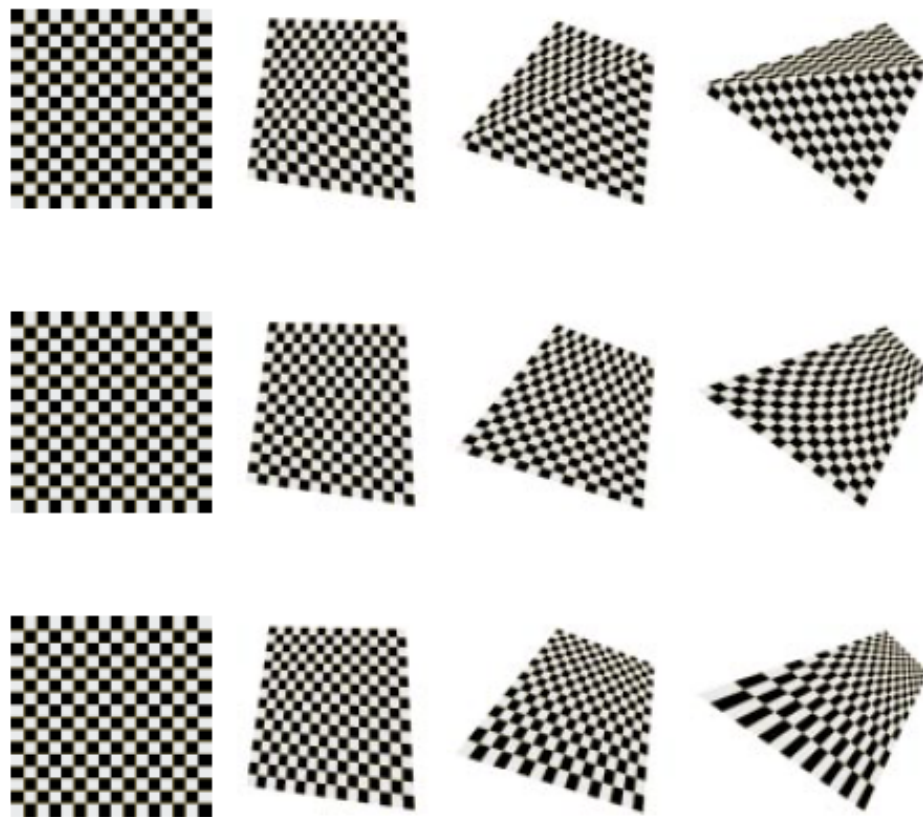


Figura 2.9: Confronto fra le tre tecniche, a partire dall'alto: affine, bilineare e proiettiva

1.2.2 Trasformazioni non parametriche

Le trasformazioni non parametriche, a differenza delle globali, non agiscono su tutti i punti dell'immagine e forniscono un controllo molto più localizzato nelle regioni su cui si vuole operare.

Questa caratteristica rende le trasformazioni non parametriche le più utilizzate nell'ambito del *morphing* tra immagini.

Anche in questa categoria esistono tre tecniche usate frequentemente nel campo del *warping*: *Mesh Warping*, *Feature Based Warping* e *Thin Plate Spline*. [4]

Mesh Warping

Il pioniere di questa tecnica è D. Smythe della Industrial Light & Magic, che lo ha utilizzato nel film “Willow” (1988).

Il *Mesh warping* è una tecnica di trasformazione 2D che si basa sulla costruzione di una *mesh* sull'immagine sorgente e di una sull'immagine finale.

Una *mesh* (maglia in italiano) è una collezione di vertici, spigoli e facce che definiscono la forma di un oggetto poliedrico nella computer grafica e nella modellazione solida.

La *mesh* costruita sull'immagine sorgente è una griglia regolare formata da poligoni solitamente di forma triangolare o quadrilatera. Modificando la posizione dei nodi creati dai poligoni si altera anche la forma di questi ultimi creando una deformazione nell'immagine (Figura 2.10).

Il cambiamento di posizione dei nodi nel *mesh warping* provoca anche un'alterazione delle posizioni di tutti i pixel vicini a questi nodi per interpolazione bilineare se si tratta di quadrilateri o di interpolazione affine se invece i poligoni sono dei triangoli.

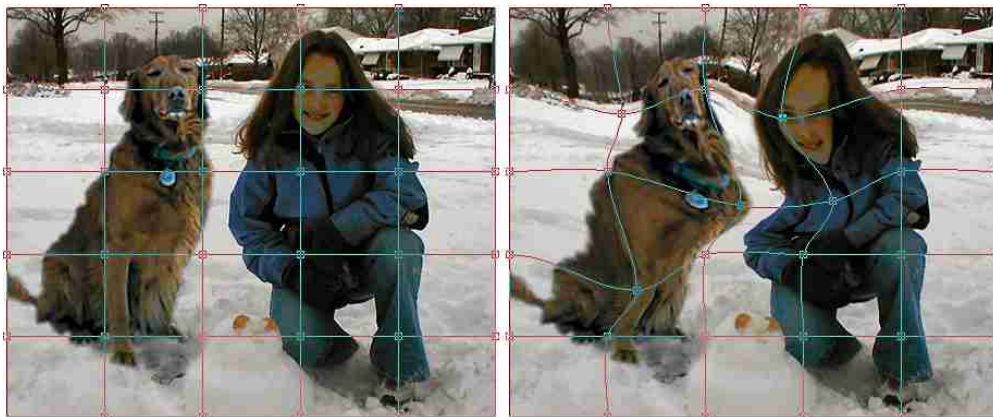


Figura 2.10: Esempio di *mesh warping*

Il processo di *Mesh Warping* fa uso di due matrici per la rappresentazione delle coordinate dei nodi per l'immagine sorgente (x,y) e le corrispondenti modificate (u,v) ; nella prima fase viene creata una nuova matrice costituita dalle coordinate (u,y)

cioè dai valori delle ascisse dei nodi dell'immagine finale e dei corrispondenti valori delle ordinate dei nodi di quella di origine.

Quindi si stabilisce il *mapping* sull'immagine tra le coordinate dei pixel di questa nuova *mesh* e quelle corrispondenti nella *mesh* originale, ottenendo così un'immagine intermedia in cui le ascisse delle coordinate dei pixel corrispondono a quelli dell'immagine finale e le ordinate a quelle dell'originale.

Nella seconda fase si procede con un'altra interpolazione questa volta tra la *mesh* intermedia creata al primo passo e quella di destinazione, ottenendo l'immagine di destinazione (Figura 2.11).

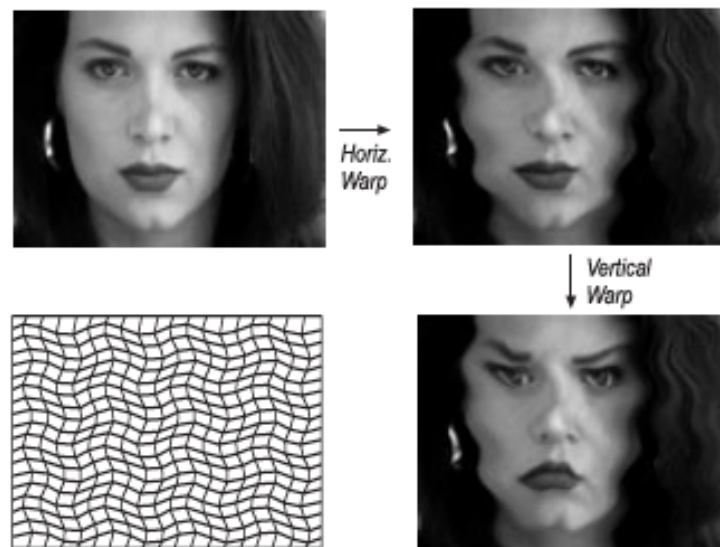


Figura 2.11: Fasi dell'algoritmo di *Mesh Warping*

Il *warping* con questa tecnica risulta essere molto rapido ed ha il costo computazionale più basso tra i tre metodi descritti, ma l'utilizzo della *mesh* con struttura regolare risulta essere scomodo se si vogliono descrivere nel dettaglio certe parti dell'immagine e in maniera più superficiale altre.

Per avere più controllo dell'immagine è necessario aggiungere nodi infittendo la *mesh*; ciò rende molto difficile la gestione dei nodi e limita la deformazione applicabile ad un nodo, a causa del fatto che durante la deformazione i nodi non si possono scambiare di posizione poiché la *mesh* di origine e di destinazione devono avere la stessa struttura e la stessa organizzazione matriciale.

Feature-Based Warping

Questa tecnica si basa su uno degli algoritmi più famosi nell'ambito del *morphing*, quello di Beier e Neely utilizzato nel video musicale “*Black & White*” di Michal Jackson.

Nel *Feature Base Warping* vengono usati segmenti orientati per marcare i tratti più rilevanti, le caratteristiche fondamentali (da qui il nome *feature*) tra l'immagine sorgente e quella di destinazione.

Ogni segmento nell'immagine sorgente corrisponde a un segmento nell'immagine finale e si calcola la trasformazione tra i due.

Tutti i punti vicini ai segmenti sono influenzati dalla trasformazione di questi e cambiano posizione a loro volta; più un punto è vicino a un segmento maggiore sarà l'influenza esercitata sulla sua posizione, viceversa se un punto è lontano da un segmento riceverà minore influenza.

Nell'immagine di origine, avendo come riferimento un segmento orientato $P'Q'$ e un punto X' , viene indicata con v la distanza di X' perpendicolare al segmento e con u la proiezione di $P'X'$ su $P'Q'$ (Figura 2.12).

Considerando $P'Q'$ come un sistema di riferimento u e v sono le coordinate di X' rispetto a questo sistema.

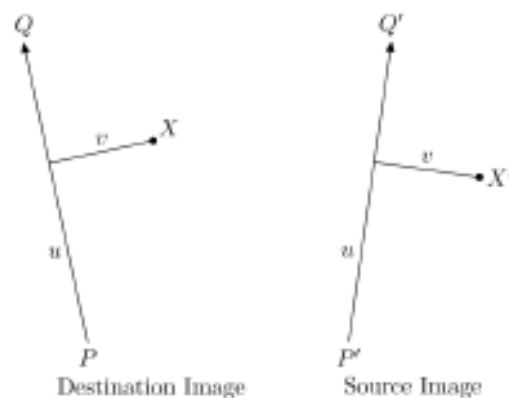


Figura 2.12: *Feature-Based Warping* con una linea

Volendo trovare le coordinate nell'immagine di destinazione di X rispetto al segmento orientato PQ , e sapendo che solo la coordinata u è normalizzata con la

lunghezza del segmento, quindi un allungamento o compressione del segmento comporta la variazione di u ma non di v , si calcola u proiettando PX su PQ :

$$u = \frac{(X - P)(Q - P)}{\|Q - P\|^2}$$

v si calcola come la proiezione di PX nel vettore perpendicolare a PQ che passa per X :

$$v = (X - P) \frac{\text{perp}(Q - P)}{\|Q - P\|}$$

Trovando u e v è possibile calcolare la posizione di X' con la seguente formula:

$$X' = P' + u(Q' - P') + \frac{v[\text{perp}(Q' - P')]}{\|Q' - P'\|}$$

Da P' ci si muove lungo $P'Q'$ della stessa misura u di cui ci eravamo mossi lungo PQ , poi ci si sposta perpendicolarmente a $P'Q'$ di una distanza pari a v .

Grazie a questo algoritmo si trasformano le coordinate di ogni pixel dall'immagine di destinazione a quella sorgente traslando, ruotando o scalando trasformando l'intera immagine.

Con una coppia di segmenti non tutte le trasformazioni affini sono possibili, in particolare le scalature uniformi e lo shearing non si possono realizzare.

Questo problema si risolve utilizzando più linee.

Si calcolano le coordinate (u_i, v_i) del punto X per ogni segmento orientato, per poi combinare i risultati per ogni X'_i trovato calcolando lo spostamento tra la locazione del pixel nell'immagine sorgente e in quella di destinazione $D_i = X'_i - X$, e ricavando la media pesata dei vari D_i ottenuti (Figura 2.13).

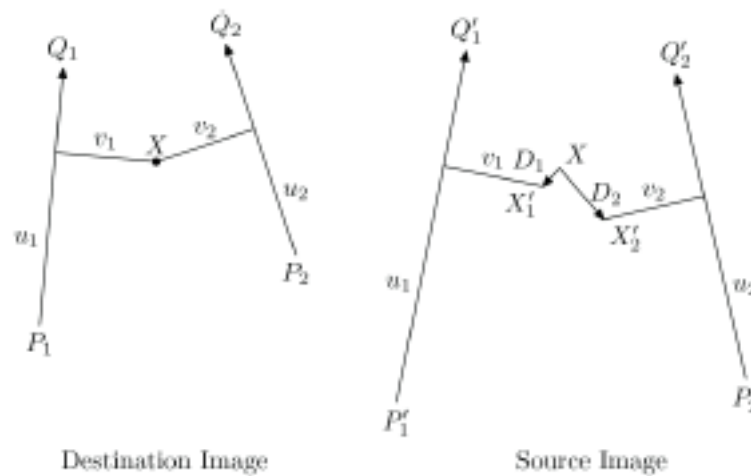


Figura 2.13: *Feature-Based Warping* con più segmenti

Come è stato menzionato in precedenza la distanza del punto dal segmento influisce sullo spostamento della posizione di questo, ciò viene determinato con una formula che stima il peso da utilizzare nella media pesata:

$$W_i = \left(\frac{\text{length}_i^p}{a + \text{dist}} \right)^b$$

Dove length è la lunghezza del segmento P_iQ_i , dist è la distanza del pixel dal segmento, a e b sono costanti che possono essere usate per cambiare l'effetto del segmento sulla media finale.

La costante a è la costante che determina quanto il segmento contribuisce ad alterare i pixel circostanti, se $a = 0$ si ha un controllo totale sui pixel, che si sposteranno assieme al segmento

La costante b determina il peso del segmento sull'intera immagine, se $b = 0$ per ogni segmento, la media degli effetti diventa aritmetica.

La costante p determina quanto contribuisce la lunghezza del segmento nel determinare il peso da associarvi, con $p = 1$ per tutti i segmenti più lunghi sono quelli con più influenza.

Una volta calcolati i pesi è possibile trovare la posizione di X' :

$$X' = X + \sum_i W_i D_i$$

Questo algoritmo porta ad ottimi risultati e risulta essere molto vantaggioso in quanto, invece di specificare e gestire un alto numero di nodi, richiede soltanto di scegliere in modo appropriato i tratti dell'immagine da marcare e di pesare a dovere i segmenti orientati.



Figura 2.14: Esempio di *Feature-Based Warping* con due linee

Purtroppo il calcolo del posizionamento di ogni pixel all'aumentare dei segmento richiede un maggiore tempo di computazione, ciò causa lentezza nell'esecuzione del *warping*.

Un altro problema riguardante questo algoritmo è la scelta dei segmenti orientati e la loro gestione dei pesi, perché l'influenza dei segmenti possono alterare lo stato o annullare gli effetti prodotti da altre linee.

Thin Plate Spline

Il metodo *Thin Plate Spline*, similmente agli algoritmi precedenti, richiede la segnalazione dei tratti fondamentali dell'immagine sorgente e dell'immagine di destinazione con dei punti di controllo.

In seguito le griglie sulle quali risiedono tali punti verranno sovrapposte avendo un risultato simile alla figura 2.15 nell'immagine di sinistra.

Per riuscire ad allineare i tratti fondamentali delle due immagini questo metodo curva la superficie delle griglie sovrapposte in modo da allineare i punti di controllo aggiunti precedentemente.

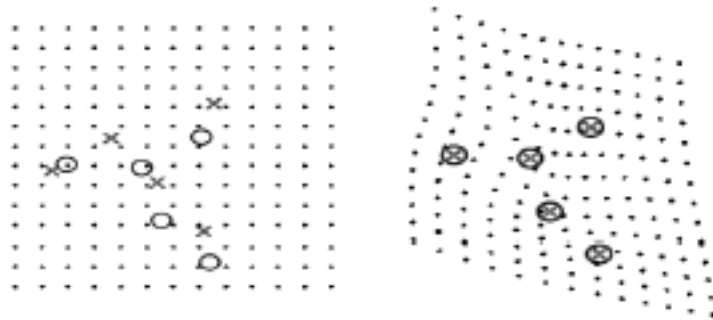


Figura 2.15: *Thin Plate Spline*

Per stabilire il *mapping* tra le due superfici si utilizza la seguente funzione interpolante:

$$u(x, y) = \sum_{i=1}^m w_i f(\sqrt{(x - x_i)^2 + (y - y_i)^2}) + a_{10}x + a_{01}y + a_{00}$$

La stima dei parametri può essere effettuata attraverso il sistema descritto di seguito in forma matriciale:

$$\begin{bmatrix} K & P \\ P^t & O \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} k \\ o \end{bmatrix}$$

dove

$$K_{ij} = f(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2})$$

la i -esima riga si P è (I, x_i, y_i) , O è una matrice 3×3 di zeri, o è un vettore colonna con tre zeri, w e k sono vettori colonna formati da w_i e u_i rispettivamente, e a è il vettore colonna con elementi a_{00} , a_{10} , a_{01} .

La stima dei parametri si ottiene risolvendo le due equazioni vettoriali del precedente sistema:

$$W = K^{-1}(k - Pa)$$

$$a = (P^t K^{-1} P)^{-1} P^t K^{-1} k$$

Il warping con *Thin Plate Spline* risulta essere molto comodo in quanto offre totale libertà all'inserimento dei punti di controllo, e con questo metodo bastano pochi punti per avere un risultato gradevole.

Questa proprietà lo rende il metodo più utilizzato nelle applicazioni di morphing convenzionali.

Per un morphing più preciso però questo algoritmo non è adatto.

Il costo computazionale di questa tecnica è molto più alto rispetto a quelli dei metodi precedenti.

1.3 Face Morphing: i sistemi automatici allo stato dell'arte

Nella maggior parte dei software di *face morphing* commerciali per realizzare una sequenza di immagini *morphed*, dopo aver effettuato il caricamento di due immagini nel programma, l'utente deve settare all'interno delle immagini la posizione dei punti di controllo (*landmarks*) o linee (a seconda dell'algoritmo implementato nel software

usato) che segnaleranno i tratti fondamentali del soggetto rappresentato nella figura, che nel caso di un volto umano saranno il contorno del viso, gli occhi, le sopracciglia, la bocca e il naso (Figura 2.16).



Figura 2.16: Esempio di posizionamento dei *landmarks*

I punti andranno posizionati a coppie, uno sulla prima figura e l'altro sulla seconda, con lo scopo di indicare la diversa posizione del tratto che si desidera marcare sulle due immagini.

Grazie a questi punti il software creerà una sequenza di *warping* dell'immagine *source* utilizzando come riferimento i punti di controllo settati nella seconda immagine, in modo tale che i tratti della prima immagine diventino in modo graduale quelli della seconda, come in figura 2.17.

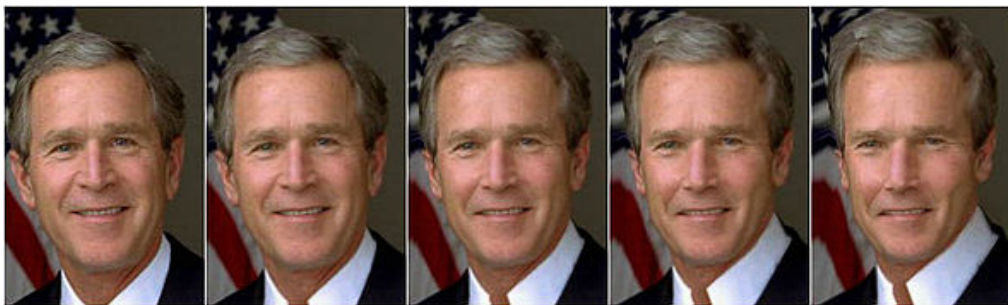


Figura 2.17: *Warping* che deforma il viso di Bush per assomigliare a quello di Schwarzenegger

Nell'altra immagine invece il programma realizzerà una sequenza inversa di *warping* rispetto alla prima immagine usando come riferimento i punti di controllo di quest'ultima (Figura 2.18).



Figura 2.18: *Warping* inverso della foto di Schwarzenegger per assomigliare a quella di Bush

La qualità del risultato finale di un *morphing* dipenderà quindi, oltre che dal tipo di software scelto, dal numero di punti di controllo usati e dal loro posizionamento.

Al termine del settaggio dei punti di controllo all'utente non è più richiesta nessuna azione, il software si occuperà della creazione delle immagini *warped*, avendo come riferimento le posizioni dei tratti fondamentali, e infine della dissolvenza incrociata.



Figura 2.19: *Face Morphing* tra la foto di Bush e quella di Schwarzenegger

L'immagine *morphed* che assomiglia ad entrambi i soggetti in egual modo è l'immagine centrale di una sequenza, la stessa immagine che viene data come risultato da un software di *morphing* che non fornisce al termine una sequenza di immagini. [3]

Capitolo 2

Riconoscimento del volto

Un sistema d'identificazione biometrico è un particolare sistema automatico in grado di identificare o di verificare l'identità di una persona sulla base di caratteristiche biologiche (es. impronta digitale, iride, mano, retina, volto, DNA) o comportamentali (es. voce, firma, stile di battitura), confrontandole con dei "modelli" precedentemente creati e memorizzati.

Il riconoscimento è un problema complesso che solitamente si basa sull'analisi di immagini statiche o sequenze video e richiede dapprima la localizzazione dei volti nelle immagini e successivamente il riconoscimento dei volti tramite confronto con un database di soggetti noti.

Il riconoscimento facciale è un'area molto attiva nella Computer Vision, già a partire dagli anni 60' gli scienziati hanno iniziato ad utilizzare i computer per riconoscere i volti umani, per realizzare infine nel 1973 il primo sistema automatico per il *face recognition* proposto da Takeo Kanade. [5]

Oggi il riconoscimento del volto viene usato in ambiti di sicurezza, robotica, HCI (*human-computer interaction*), camere digitali, giochi e intrattenimento.

Da un punto di vista biometrico, il volto non è caratterizzato da un'elevata permanenza: le molteplici espressioni del volto, l'età, i radicali cambiamenti fisici, la presenza di occhiali, sono esempi di caratteri esteriori che possono mutare nel tempo rendendo difficoltoso il riconoscimento facciale. Le caratteristiche "non permanenti" del volto, implicano una notevole complessità di problemi tecnici da risolvere. Ciò

nonostante, sono state sviluppate con successo alcune tecniche che consentono di conseguire risultati soddisfacenti d'identificazione personale.

Il processo di riconoscimento da parte di un sistema avviene secondo le seguenti fasi:

- Fase di *pre-elaborazione*: garantisce che l'immagine a cui sarà applicato il processo di riconoscimento soddisfi dei requisiti standard stabiliti in precedenza, per esempio se si vuole avere il soggetto al centro dell'immagine, che lo sfondo sia ben distinto dal volto della persona, che la luminosità sia ottimale per l'acquisizione dell'immagine, o se si desidera che il soggetto si trovi ad una certa distanza dall'obiettivo.

La maggior parte delle volte questi controlli sono effettuati dalle apparecchiature utilizzate per il rilevamento delle immagini, un esempio sono i sensori di rilevamento appartenenti alle telecamere di videosorveglianza.

- Fase di *face detection*: consiste nella localizzazione dei volti delle persone presenti nelle immagini.

La localizzazione del volto nelle immagini può risultare un compito complesso.

Per esempio quando il riconoscimento si deve effettuare su un video, quindi una sequenza di fotogrammi, dove non è possibile controllare fattori come sfondo, illuminazione, posa del soggetto e distanza dall'obiettivo, per cui il rilevamento del volto si mostra più complicato, anche se c'è la possibilità di effettuare una scelta fra i diversi fotogrammi che formano il video.

Quando l'immagine da elaborare è prelevata in un ambiente dove sono controllati i fattori citati precedentemente la localizzazione del volto è relativamente semplice.

- Fase di estrazione delle *feature*: L'estrazione di un particolare (*feature*) dal volto rilevato nella fase precedente è l'individuazione e rappresentazione in forma matematica delle caratteristiche fondamentali di un volto come per esempio gli occhi, la bocca, i capelli, il naso o i lineamenti. Questa

rappresentazione dei dati è utile in quanto rende possibile l'identificazione del volto e la distinzione dai volti di altre persone.

- Fase di *riconoscimento*: grazie alla rappresentazione dei particolari di un viso in forma matematica è possibile effettuare un confronto tramite degli algoritmi tra questi dati e le informazioni rilevate da un'altra immagine del soggetto che si vuole identificare.

In seguito saranno descritte queste fasi nel dettaglio e saranno illustrati gli algoritmi di riconoscimento più utilizzati.

2.1 Face Detection

Un sistema di *face detection* riceve in input l'immagine digitale in cui si vogliono localizzare dei volti e restituisce in output l'immagine con le facce individuate (Figura 3.1).

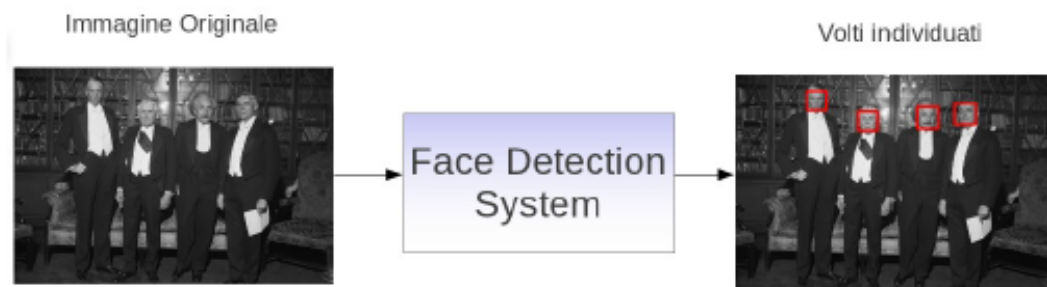


Figura 3.1: Esempio di sistema di localizzazione di volti

Per riuscire a determinare la posizione di un volto in un'immagine è utile servirsi degli algoritmi di rivelazione di zone di pelle in modo da poter estrarre delle sotto-aree che contengono pixel classificabili in base al colore, simile a quello della pelle umana, con lo scopo di restringere sensibilmente il campo per la ricerca della presenza di un volto. Una volta individuate le aree d'interesse, le tecniche più

utilizzate al momento per la localizzazione del volto si basano su un'analisi dell'immagine tramite una finestra mobile che scorre sull'immagine; ciascuna finestra viene poi rappresentata con opportune *feature* numeriche e classificata da un sistema esperto come “volto” o “non volto”.

2.1.1 Skin Segmentation

Sulla base degli studi di Brand e Mason [6] che hanno sperimentato numerose tecniche per l'individuazione della pelle in un'immagine, Igawa [28] riuscì a capire che considerando il rapporto fra il livello di rosso (R) e il livello di verde (G) presenti nella pelle umana il colore rosso è sempre presente in modo significativo e normalizzando questo valore (dividendolo per il livello del verde) si può ottenere una buona indicazione della presenza di pelle.

Wark in seguito [6] mostrò che filtrare i pixel di un'immagine a seconda del rapporto R/G, accettando solo quelli tali che:

$$L_i < R/G < L_s$$

poteva essere un buon metodo per individuare la pelle umana.

In seguito agli esperimenti di Brand e Mason su oltre 12000 immagini, alcune contenenti pelle umana e altre no, si scoprì che il rapporto R/G di pixel color pelle in un grafico avente come ascisse il colore verde e come ordinate il colore rosso è ristretto ad un intervallo con limite inferiore pari a 1 a differenza dei pixel colore “non-pelle” (Figura 3.2).

Questi esperimenti dimostrarono che il 94.7% dei pixel provenienti da immagini di pelle umana viene accettato, ma che insieme ad esso anche il 35.6% dei pixel proveniente da pixel non contenenti pelle. [6]

Nonostante ciò la velocità e l'efficacia di questa tecnica lo rendono un ottimo metodo per restringere il campo di ricerca di un volto all'interno di un'immagine.

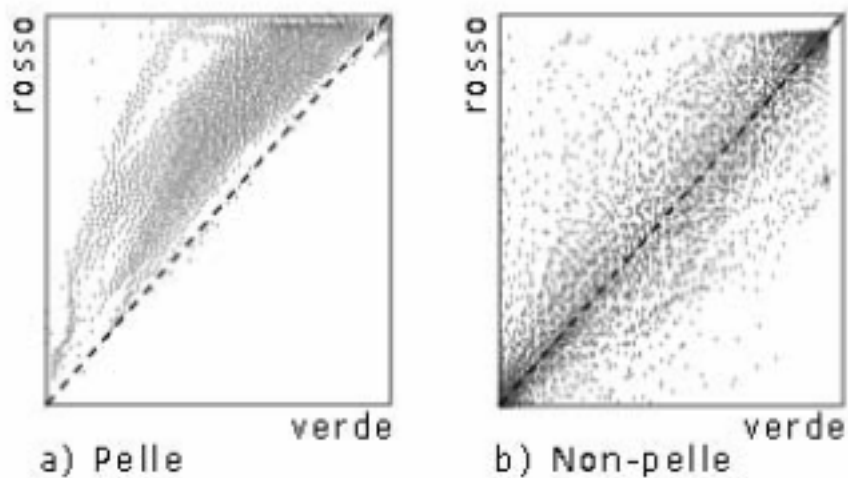


Figura 3.2: Grafici del livello di rosso vs. il livello di verde per: a) immagini di pelle b) immagini di non-pelle

2.1.2 Il metodo Viola-Jones

Uno degli algoritmi più efficienti e di gran lunga più utilizzati per la localizzazione del volto è il localizzatore di *Viola-Jones*. [7]

Questo algoritmo classifica delle aree di pixel basandosi sull'utilizzo di semplici *feature* chiamate *Haar-like* per la similitudine con le *wavelet di Haar*. [8]

Una *feature Haar-like* è costituita da due o più regioni rettangolari adiacenti di pixel dalle quali è estratta un'informazione numerica (Figura 3.3).

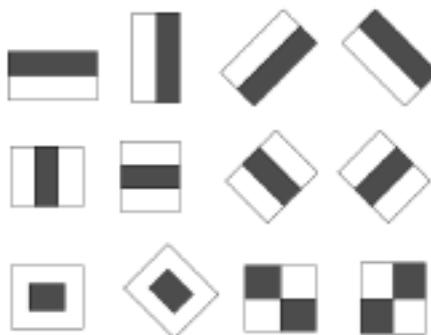


Figura 3.3: *Feature Haar-like*

Come si può vedere dalle immagini le *feature* sono composte da due o più regioni di colore bianco o nero e l'informazione che ne risulta è la differenza tra il valore della *feature* dato dalla somma dell'intensità dei pixel $p_b(i)$ collocati all'interno dell'area in nero, e la somma dell'intensità dei pixel $p_w(i)$ nell'area bianca.

Le *feature* caratterizzano così il contrasto locale delle zone salienti dell'immagine.

Per migliorare l'efficienza del sistema nel calcolo delle *feature*, invece di calcolare ogni volta la somma dei pixel all'interno di ciascuna regione, viene utilizzata un'immagine ausiliaria chiamata *immagine integrale* su cui si eseguiranno tutte le operazioni.

L'immagine integrale ha le stesse dimensioni dell'immagine originale, ma i valori d'intensità di ogni pixel sono calcolati come la somma del pixel in questione e di tutti i pixel alla sua sinistra e sopra di esso (Figura 3.4).

In questo modo il calcolo della somma dei pixel, all'interno di un'area, è realizzabile in un tempo costante, semplicemente facendo quattro accessi alla memoria e tre operazioni di somma.

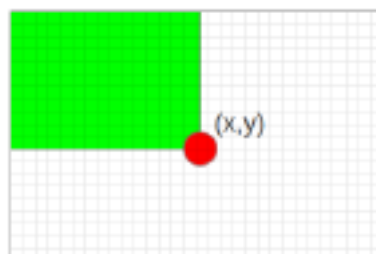


Figura 3.4: Pixel presi in considerazione per il calcolo dei nuovi pixel dell'*immagine integrale*

L'algoritmo di classificazione delle finestre in "volto" o "non volto" utilizza una variante di *AdaBoost* [9] sia per selezionare le *feature* che per addestrare il *classificatore*.

Un classificatore è un modello sulla base del quale si decide di quale gruppo o classe appartiene un determinato oggetto con l'aiuto della combinazione lineare delle sue caratteristiche.

Nella forma originale AdaBoost unisce tra loro una serie di funzioni di classificatori “deboli” (*weak classifiers*) che cercheranno di eliminare le sotto-finestre che certamente non conterranno l’oggetto in esame ottenendo un classificatore finale sufficientemente robusto detto *strong classifier (boosting)*.

Se il classificatore addestrato non trova un oggetto che invece è presente oppure ne indica erroneamente la presenza, si può ricalibrare il suo addestramento aggiungendo gli esempi corrispondenti (positivi o negativi che siano) al suo *training set*.

In pratica, ciascun singolo classificatore viene ristretto all’insieme delle funzioni di classificazioni che dipendono ciascuna da una singola *feature*.

L’algoritmo “debole” di classificazione è quindi progettato per scegliere la singola *feature* rettangolare che meglio separa gli esempi positivi dagli esempi negativi. Per ogni *feature*, il classificatore debole determina la soglia ottimale della funzione di classificazione, per la quale il minor numero di esempi viene classificato in modo errato. Per determinare lo *StrongClassifier* è necessario definire e normalizzare dei pesi (*weights*) che si calcolano a partire dal numero di esempi positivi e negativi a disposizione: essi infatti permettono di calcolare il miglior valore di soglia (*threshold*) che minimizza l’errore di classificazione per ciascuna *feature*. La *feature* scelta per il classificatore corrente è quella che permette di ottenere l’errore di classificazione complessivamente più basso. Ad ogni passo di *boosting* viene creato un classificatore base (*weak*). Per riconoscere la presenza di un oggetto conosciuto in un’immagine arbitraria bisognerebbe raccogliere tutti questi classificatori base e sommarli per crearne uno che sia valido per il riconoscimento effettivo dell’oggetto, utilizzando questo classificatore generale (“*strongClassifier*”) per verificarne la presenza in tutte le possibili sotto-finestre dell’immagine da analizzare. Procedendo in questo modo, il costo computazionale diviene visibilmente proibitivo.

Viola e Jones hanno risolto questo problema con l’utilizzo di una struttura a cascata di classificatori via via più complessi invece del classificatore generale.

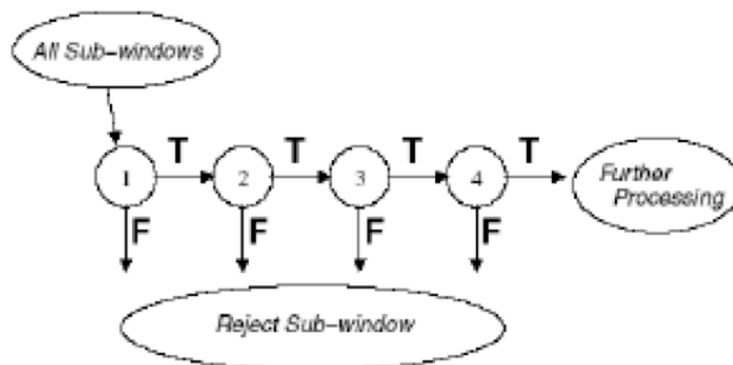


Figura 3.5: Cascata di classificatori

Una serie di classificatori vengono applicati a tutte le sotto-finestre presenti nell'immagine (Figura 3.5). Per ognuna di queste, un risultato positivo del primo classificatore conduce all'analisi del secondo classificatore, che in caso di risultato positivo conduce all'analisi di un terzo classificatore e così via, eliminando la sotto-finestra corrente alla prima occorrenza di un risultato negativo.

In questo modo le risorse verranno consumate solo dalle aree che potrebbero condurre ad una individuazione dell'oggetto cercato, riducendo drasticamente il numero di ricerche nell'immagine.

2.2 Estrazione delle feature per il riconoscimento

Quando si deve eseguire un algoritmo su un ampio insieme di dati su cui si sospetta ridondanza e dispersione, provocando un alto costo computazionale la riduzione di dimensionalità dello spazio dei dati è utile per diminuire il numero di dati su cui lavorare sintetizzando e selezionando le informazioni principali.

L'analisi del contenuto di un'immagine non avviene generalmente a partire dai pixel che la caratterizzano ma sulla base di un insieme ristretto di informazioni numeriche.

Tale elaborazione è nota come estrazione delle *feature* e ha lo scopo di estrarre dall'immagine una rappresentazione ridotta, codificata tramite un insieme limitato di dati numerici, che catturi le caratteristiche salienti delle immagini. Ad esempio, nel caso del riconoscimento del volto possono essere estrapolate le informazioni riferite agli elementi di maggior rilevanza del viso (occhi, naso, bocca, fronte, lineamenti), sulle quali viene costruito un modello matematico di rappresentazione. Tale insieme ridotto di informazioni può essere ottenuto applicando, singolarmente o in combinazione, tecniche di estrazione di *feature* e tecniche di riduzione di dimensionalità. Nei seguenti paragrafi saranno descritte dapprima le più note tecniche di riduzione di dimensionalità dei dati e, successivamente, alcune delle tecniche di estrazione di *feature* più utilizzate per il riconoscimento del volto. Va inteso che a fronte di *feature* estratte utilizzando queste ultime tecniche è poi comunque possibile operare un'ulteriore riduzione ricorrendo a tecniche di riduzione di dimensionalità.

2.2.1 Dimensionality Reduction

Obiettivo dei metodi per la riduzione di dimensionalità (*dimensionality reduction*) è quello di eseguire un *mapping* dallo spazio iniziale R_d ad uno di dimensione inferiore. Le principali tecniche sono: PCA (*Principal Component Analysis*), LDA (*Linear Discriminant Analysis*) e ICA (*Independent Component Analysis*). [5]

Questi metodi lineari di analisi statistica sono utilizzati per permettere di trovare patterns all'interno di un insieme di dati di grandi dimensioni, classificando tali dati in gruppi in base al loro grado di similarità.

PCA

La trasformata di Karhunen-Loeve (KL), nota anche col nome PCA, esegue un *mapping* lineare con lo scopo di preservare al massimo l'informazione dei dati nello spazio originale. Lo spazio ridotto viene creato da delle nuove variabili ottenute come combinazione lineare delle variabili originali, in modo che un numero ridotto

di queste nuove variabili siano in grado di spiegare una porzione della varianza totale dei dati. Vengono calcolati gli autovettori della matrice di covarianza e solo gli autovettori corrispondenti ai più alti valori di varianza vengono conservati, limitando così l'analisi (per varianza) alle principali variabili.

LDA

Anche LDA, altrimenti noto come Fisher Discriminant Analysis, opera un mapping lineare verso uno spazio a dimensione ridotta. La differenza fondamentale rispetto a KL sta nel fatto che mentre KL privilegia le dimensioni che al meglio codificano e rappresentano i pattern, LDA privilegia le dimensioni che li discriminano al meglio.

ICA

ICA è una generalizzazione di PCA, che tende a trovare una trasformazione lineare così che i componenti dello spazio risultante siano più indipendenti possibile.

Gli autori di [29] hanno proposto un metodo per il riconoscimento del volto basato sulla rappresentazione ICA. Essi hanno osservato che in PCA il primo componente principale dell'immagine del volto è strettamente correlato alla posa e alle condizioni di illuminazione e che rimuovendo questa componente principale dalla rappresentazione tramite PCA le prestazioni dell'algoritmo miglioravano. Infatti hanno poi adottato una tecnica per rimuovere questo componente prima di applicare ICA e i risultati sperimentali hanno dimostrato che ICA ha un maggior potere di discriminazione delle classi rispetto a PCA nel caso di riconoscimento di volti in presenza di cambiamenti di posa ed illuminazione.

2.2.2 Filtri di Gabor

I *filtri di Gabor* vengono spesso utilizzati nell'estrazione delle caratteristiche, grazie alle loro proprietà di localizzazione sia nel dominio del tempo che nel dominio della frequenza. [10]

Questi filtri sono in grado di estrarre le *texture* (tessiture) di un'immagine, cioè le proprietà che descrivono la distribuzione spaziale dei livelli di grigio che corrispondono ai pixel di una determinata regione.

Un filtro di Gabor (Figura 3.6c) è un filtro lineare ottenuto mediante la modulazione di una funzione Sinusoidale (Figura 3.6a) con una funzione Gaussiana di deviazione standard " Φ " (Figura 3.6b). Il filtro, costituito da una parte reale e una immaginaria, è regolato da 3 parametri:

- la frequenza f della sinusoide
- l'orientazione θ della sinusoide rispetto al piano x,y
- l'ampiezza Φ della gaussiana

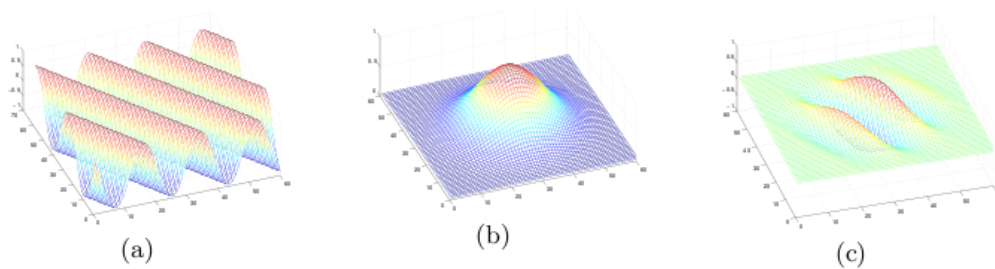


Figura 3.6: (a) funzione sinusoidale, (b) kernel Gaussiano, (c) filtro di Gabor

Dato il filtro di Gabor, $g(x, y, \theta, \Phi)$ centrato nell'origine; x e y rappresentano le coordinate spaziali dell'immagine ed i parametri Φ e θ rispettivamente la scala e l'orientazione. Di seguito si presenta la formula del filtro:

$$g(x, y, \theta, \phi) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \exp(2\pi\theta i(x \cos \phi + y \sin \phi))$$

Il parametro Φ è la frequenza spaziale della funzione sinusoidale, la cui unità di misura sono i pixels. Il valore θ specifica l'orientazione della funzione sinusoidale, misurato in gradi.

Il risultato dell'equazione è un filtro di Gabor visualizzato come due immagini rappresentanti la parte reale e la parte immaginaria del filtro (Figura 3.7). Il filtro viene creato con l'origine al centro dell'immagine.



Figura 3.7: Filtro di Gabor, a sinistra la parte reale e a destra la parte immaginaria del filtro

2.3 Riconoscimento delle feature

L'ultima fase del processo di *face recognition* è il riconoscimento delle *feature* corrispondenti a due immagini differenti.

Questa fase si avvale oltre di algoritmi complessi di riconoscimento, anche di una conoscenza acquistata durante un addestramento del classificatore, sulla base di un insieme di dati detto *training set*.

Con *training set* si identifica un insieme di dati usato per addestrare un sistema supervisionato (come una rete neurale o un classificatore probabilistico), in modo tale che il sistema ad ogni test dell'addestramento apprenda la capacità di

determinare le differenti categorie di appartenenza dei dati in base alle loro caratteristiche (*feature*).

In seguito vengono analizzati i metodi più noti in letteratura che costituiscono un punto di riferimento fondamentale per i software di riconoscimento del volto.

2.3.1 Eigenfaces

Il metodo *Eigenfaces*, conosciuto come “metodo delle autofacce”, è stato sviluppato nel 1991 da Matthew Turk del Computer Science Department of University of California e da Alex Pentland del Mit Media Laboratory, a partire dai progetti creati da Kirby e Sirovich nel 1987 che dimostravano che l’Analisi delle Componenti Principali poteva essere usata per la rappresentazione delle *feature* fondamentali di un volto. [11] [5]

Questo metodo infatti si basa sull’estrazione delle *feature* grazie all’utilizzo della procedura PCA, da cui ne risulteranno degli autovettori chiamati *eigenfaces* usati come base del nuovo spazio.

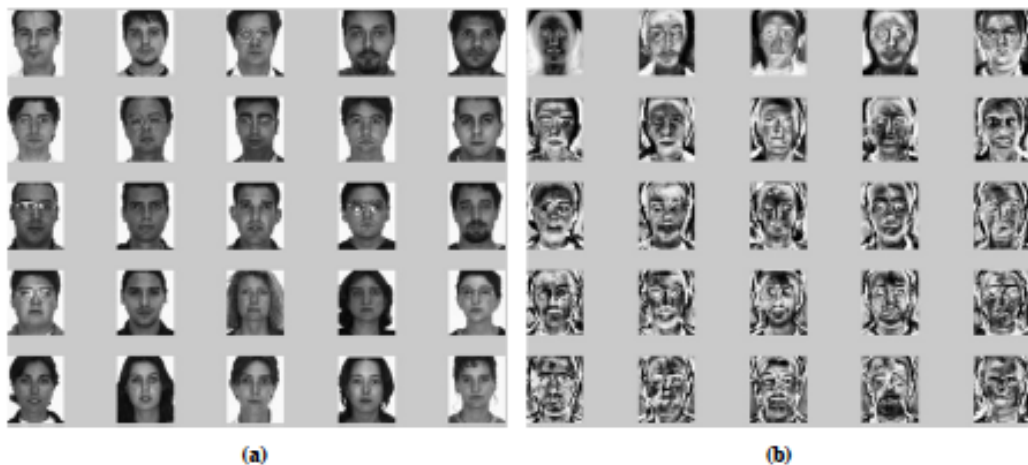


Figura 3.8: Esempio di facce utilizzate in un training set (a) e le relative *eigenfaces* (b)

Ogni *eigenface* descrive una certa caratteristica come la linea dei capelli, la simmetria, la larghezza del naso o i contorni di quelle componenti non chiaramente distinguibili, possono essere pensati come una serie di *feature* che insieme

caratterizzano le differenze tra le varie immagini delle facce. La faccia di ciascuna persona può essere rappresentata esattamente come la combinazione lineare di *eigenfaces*; inoltre tale faccia può anche essere approssimata dalle sole *eigenfaces* più significative. Ne segue che ciascuna persona può essere identificata da un insieme di pesi relativi alle *feature*, ovvero può essere descritta tramite una rappresentazione molto più compatta di un'immagine.

Se consideriamo, ad esempio, immagini 100x100 pixel avremo 10000 possibili componenti ma in realtà la maggior parte degli individui può essere riconosciuto con un numero tra le 100 e le 150 *eigenfaces*.

Si supponga che l'immagine di una faccia sia rappresentata da un array bidimensionale $N \times N$ di valori a 8 bit di profondità. Un'immagine può anche essere considerata come un vettore di dimensione N^2 : ad esempio un'immagine di dimensione 256 per 256 può venire rappresentata da un vettore a 65536 dimensioni o, in modo equivalente, da un punto in uno spazio a 65536 dimensioni. Un insieme di immagini può quindi essere pensato come un insieme di punti in questo enorme spazio vettoriale. Le immagini delle facce tuttavia, avendo tutte una configurazione simile tra loro, non sono distribuite uniformemente in tale spazio e possono pertanto essere descritte da un sottospazio avente un numero inferiore di dimensioni. L'idea dell'analisi delle componenti principali è proprio quella di trovare quei vettori che rappresentano meglio la distribuzione delle immagini delle facce all'interno dell'intero spazio delle immagini. Tali vettori definiscono un sottospazio all'interno di quest'ultimo, che viene chiamato "spazio delle facce".

La complessità dell'algoritmo nella fase di training è $O(n^2 m^2 L)$ e nella fase di test di $O(LMN)$, con n e m il numero di righe e colonne della matrice immagine, L il numero di vettori proiezione, M elementi di test e N elementi di training.

La complessità dello spazio è $O(n^2 m^2)$.

Informalmente, *eigenfaces* può essere considerato un insieme di "*standardized face ingredients*", derivata da analisi statistica di molte immagini di volti. Ogni volto umano può essere considerato come una combinazione di questi volti standard.

Ecco il processo per la creazione delle *eigenfaces*:

1. preparare un training set di immagini di volti, rappresentati in una matrice T in cui ogni riga è un'immagine.
2. Calcolare e sottrarre l'immagine media ad ogni immagine originale in T .
3. calcolare gli autovettori (*eigenface*) e autovalori della matrice di covarianza. Gli autovalori sono le direzioni in cui le immagini differiscono dalle immagini media.
4. Scegliere le componenti principali. La matrice di covarianza si tradurrà in autovettori, tra i quali quelli con il più alto autovalore associato saranno tenuti.

Il riconoscimento facciale è stato la fonte di motivazione alla base della creazione di *Eigenfaces*. Purtroppo, questo tipo di riconoscimento facciale presenta alcuni aspetti negativi legati a scarsa robustezza rispetto a cambiamenti di posa o illuminazione. Perché il sistema funzioni bene, le facce devono essere viste da una visione frontale e con illuminazione simile.

2.3.2 Support Vector Machines

Le *Support Vector Machines* sono un insieme di metodi di apprendimento supervisionato per la regressione e la classificazione di pattern, sviluppati negli anni '90 da Vladimir Vapnik ed il suo team presso i laboratori Bell AT&T. [12]

Partendo da due classi di *pattern* multidimensionali SVM determina un iper-piano in grado di classificare correttamente pattern di entrambe le classi con il maggior margine possibile.

Se le classi non sono linearmente separabili, viene eseguito a priori un *mapping* dei *pattern* su di uno spazio di dimensione superiore, dove i maggiori gradi di libertà permettono di separare le classi.

Anche se SVM è stato creato come classificatore binario, esistono approcci che risolvono il problema di classificazione con più classi, come nel riconoscimento del volto, anche se questi risultano essere computazionalmente più onerosi rispetto a problemi con due classi.

Esistono due tecniche principali per la decomposizione di problemi multi-classe in più problemi binari: *one-vs-all* e *pairwise*.

Nel metodo *one-vs-all* per risolvere problemi di classificazione ad N classi, si costruiscono N classificatori binari. L' i -esimo classificatore SVM viene addestrato impostando l'etichetta della classe i -esima come positiva, ed il restante insieme di esempi con etichetta negativa. In fase di riconoscimento, si sottopone l'esempio di test a tutti gli N classificatori SVM e si etichetta con la classe il quale SVM presenta la risposta maggiormente positiva. Lo svantaggio di questo metodo è la complessità nella fase di training, proporzionale alla grandezza del numero di esempi, poiché ciascuno degli N classificatori viene addestrato usando tutti i possibili esempi.

A differenza della tecnica *one-vs-all*, *pairwise* prevede l'utilizzo di un classificatore SVM binario per ciascuna coppia di classi, si costruiscono quindi $N(N-1)/2$ classificatori. Ogni classificatore viene addestrato usando gli esempi della prima classe come positivi, e quelli della seconda come negativi. Per combinare i risultati di tutti questi classificatori si adotta la politica del massimo valore positivo, ovvero si seleziona la classe che riceve il valore massimo tra tutti gli $N(N-1)/2$ classificatori.

Il vantaggio di questa tecnica rispetto alla precedente è che il numero di esempi utilizzati per il training di ciascun classificatore è piccolo, poiché soltanto due delle originali N classi vengono considerate per ciascun addestramento, portando così a tempi di addestramento notevolmente minori.

Lo svantaggio, invece, è che ciascun esempio di test deve essere confrontato con grande numero di classificatori ($N(N-1)/2$). Questo elevato numero di confronti rallenta il test degli esempi al crescere al numero di classi N . [13]

2.3.3 Reti Neurali

In natura una rete neurale è costituita da neuroni che ricevono in input segnali chimici attraverso connessioni dette sinaptiche. I segnali chimici vengono convertiti dal neurone in impulsi elettrici (sull'assone) che sono a loro volta in grado di eccitare altri neuroni a seguito del rilascio di altri segnali chimici (l'assone termina con una connessione sinaptica).

Una rete neurale artificiale è un modello matematico che rappresenta l'interconnessione tra elementi definiti neuroni artificiali, ossia costruiti matematici che in qualche misura imitano le proprietà dei neuroni viventi. Questi modelli

matematici possono essere utilizzati sia per ottenere una comprensione delle reti neurali biologiche, ma ancor di più per risolvere problemi ingegneristici di intelligenza artificiale come quelli che si pongono in diversi ambiti tecnologici (in elettronica, informatica, simulazione, e altre discipline).

Il neurone artificiale è un modello matematico che calcola una funzione, detta funzione di attivazione. Gli ingressi di tale funzione modellano gli stimoli che il neurone biologico riceve dagli altri neuroni, mentre il risultato calcolato (uscita) descrive il segnale trasmesso dal neurone lungo l'assone. Quindi, la funzione di attivazione del neurone artificiale trasforma i valori dell'insieme degli ingressi (spazio degli ingressi) in corrispondenti valori nell'insieme delle uscite (spazio delle uscite).

Il primo modello formale (matematico) di neurone è stato proposto da McCulloch e Pitts nel 1943, ben prima che i calcolatori digitali si diffondessero. La sua funzione di attivazione può assumere due valori: neurone attivo, 1, o silente, 0. Anche i pesi sinaptici, che modellano la forza della connessione tra una coppia di neuroni possono avere solo due valori: eccitante, 1, o inibitorio, -1.

Il neurone di McCulloch e Pitts esegue la somma pesata degli stati dei neuroni a esso connessi, e, a seconda che tale valore sia maggiore o minore della soglia, assume lo stato attivo o passivo.

I neuroni sono connessi tra loro in base ai modelli utilizzati, esistono tre tipi di modelli principali: le reti di *Hopfield*, le reti *feed-forward* e le reti *ricorrenti*.

Le reti di *Hopfield* sono reti completamente connesse: ogni neurone è connesso a ogni altro neurone della rete.

Poiché la funzione di attivazione dei neuroni di una rete di *Hopfield* è di tipo bipolare, una tale rete con n neuroni è descritta dall'insieme delle soglie, e dall'insieme dei pesi sinaptici. L'uscita del neurone in un certo istante di tempo è detta stato del neurone; l'insieme degli stati dei neuroni che compongono la rete definisce lo stato della rete (Figura 3.9).

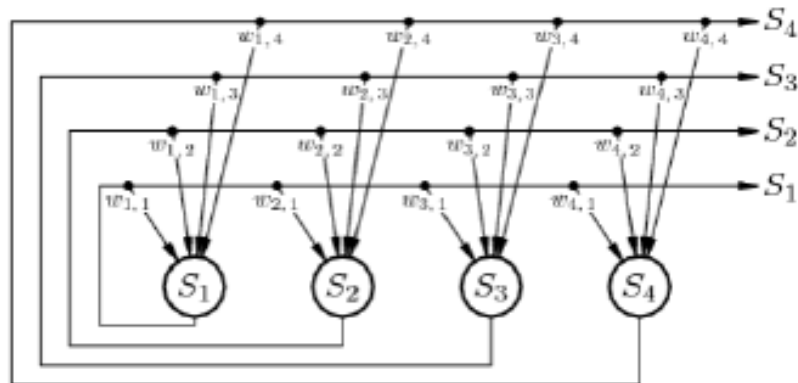


Figura 3.9: Esempio di rete di Hopfield

Le reti *feed-forward* multistrato sono così chiamate, perché il flusso di informazioni tra i neuroni procede in modo unidirezionale. Si tratta quindi di reti parzialmente connesse nelle quali i neuroni sono organizzati in sottoinsiemi, detti strati (*layer*). Gli strati sono ordinati e solo i neuroni di due strati consecutivi sono direttamente connessi. Il primo strato (quello che riceve gli stimoli dall'esterno) viene detto strato di ingresso, mentre l'ultimo strato (quello che fornisce la risposta della rete) viene detto strato di uscita (Figura 3.10). Gli strati intermedi vengono detti strati nascosti (*hidden layer*).

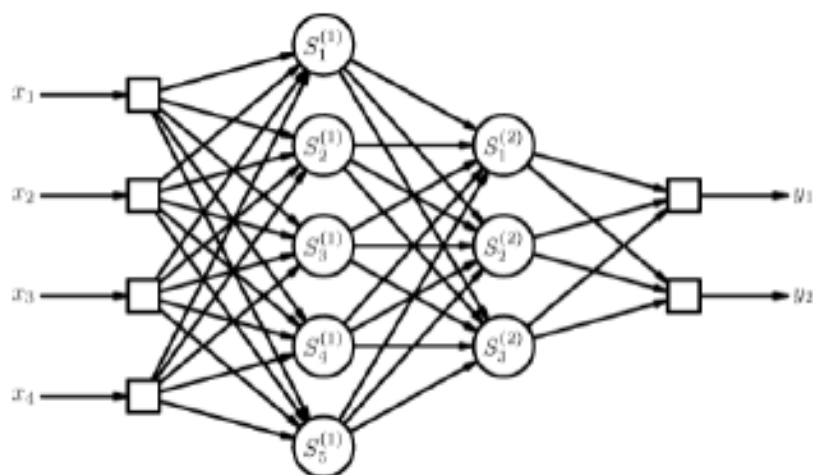


Figura 3.10: Esempio di rete feed-forward

Le reti *ricorrenti* sono reti multistrato parzialmente connesse in cui le risposte di uno o più strati possono essere conservate e fornite come ingresso negli istanti successivi. Il comportamento di queste reti è quindi dinamico: l'uscita fornita da una rete a un certo istante non dipende solo dallo stimolo fornito in ingresso all'istante considerato, ma anche dalla storia passata (Figura 3.11).

La conservazione delle informazioni degli istanti precedenti è affidata a connessioni con ritardo temporale (retroazione). A seconda del numero di tali connessioni (eventualmente con differenti ritardi), è possibile far considerare alla rete gli effetti di istanti precedenti.

Le reti ricorrenti sono chiaramente più complesse delle reti *feed-forward* multistrato, ma sono capaci di modellare sistemi con comportamento dinamico.

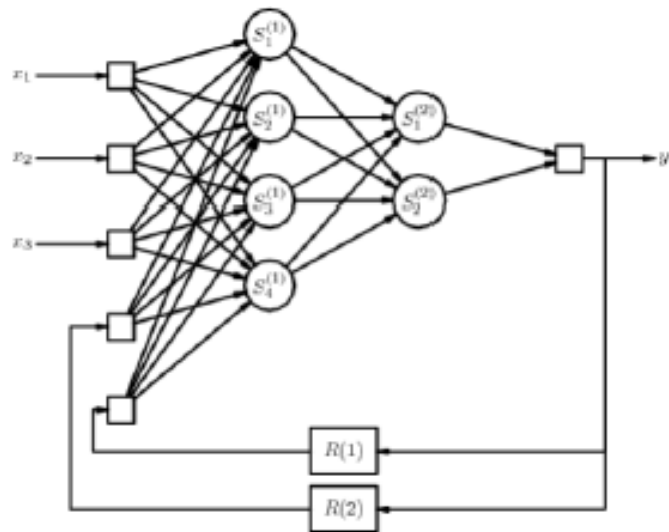


Figura 3.11: Esempio di rete ricorrente

Sono possibili due tipi di addestramento di una rete neurale: apprendimento supervisionato e apprendimento non supervisionato.

Nel caso di allenamento supervisionato vengono sottoposti alla rete, oltre ai pattern del *training set*, le risposte corrette che la rete deve generare nel caso che, durante la fase di apprendimento, questi *pattern* siano forniti in input. Queste risposte sono stringhe che codificano la classe di appartenenza: nel caso dei volti le varie classi rappresentano le identità delle persone le cui immagini sono elaborate. Al contrario,

nell'allenamento non supervisionato non vengono fornite informazioni aggiuntive alla rete oltre ai pattern del training set. La rete compie così un'autorganizzazione solamente in base alle caratteristiche strutturali dei *patterns*.

La capacità di apprendimento e di classificazione di queste reti viene sfruttata per interpretare possibili variazioni di scala e orientamento, ma anche per riconoscere soggetti deformati o parzialmente occultati.

Un metodo efficace di reti neurali per riconoscere dei soggetti tramite delle immagini campione è l'addestramento di una rete *backpropagation*.

A ogni passo di apprendimento l'algoritmo *backpropagation* calcola la relativa uscita prodotta dalla rete, e la confronta con il valore di uscita atteso. La differenza tra il valore di uscita desiderata e il valore di risposta della rete neurale costituisce l'errore commesso dalla rete stessa. Procedendo a ritroso dall'uscita della rete verso i neuroni più interni, si calcola il gradiente dell'errore rispetto ai parametri dei neuroni considerati e lo si utilizza per modificare i parametri stessi in modo da far diminuire l'errore. Questo procedimento risulta però estremamente pesante dal punto di vista computazionale, anche utilizzando piccole immagini a bassa risoluzione.

Per questo nella pratica sono state utilizzate più reti neurali opportunamente combinate.

Possono essere gestite due reti: una per l'estrazione delle *feature* (*autoassociation net*), ed una per la classificazione (*classification net*). Il ruolo della prima rete è quello di fornire alla seconda la migliore approssimazione per ognuna delle immagini in ingresso; a tale scopo viene predisposto in essa un numero sufficiente di neuroni nascosti (in genere questo numero è notevolmente inferiore al numero di immagini gestibili dalla rete).

Due reti *hyper back-propagation* indipendenti possono lavorare su vettori di dimensione limitata, costituiti da valori ricavati da alcune caratteristiche rilevanti del volto (es. naso, bocca, mento, sopracciglia, ecc.). Il risultato finale viene determinato scegliendo il migliore tra i risultati forniti dalle due reti.

Le reti neurali tradizionali mostrano da un lato una rapida attività dinamica e dall'altro una pesatura sinaptica soggetta ad un lento adattamento.

Dato che le costanti di tempo sono così diverse, le attività dinamiche possono essere trattate in modo indipendente da quelle di pesatura sinaptica. In questa ottica

vengono proposte particolari reti neurali in grado di mutare rapidamente il proprio stato durante la fase di apprendimento. Viene introdotto un concetto di correlazione temporale, che permette di esprimere espressamente relazioni tra segnali all'interno della rete. I neuroni che elaborano informazioni "legate" agiscono in modo sincrono, mentre gli altri lavorano in modo correlato (questa costituisce una decisa soluzione di "binding" della rete). Inoltre l'utilizzo di link dinamici sulle sinapsi di scambio consente, durante l'elaborazione di un segnale, una rapida riconfigurazione dell'architettura di rete. Questa riconfigurazione sinaptica fa parte dell'elaborazione del segnale, e non va confusa con il processo di addestramento della rete.

I volti vengono memorizzati come grafi etichettati che vengono rappresentati da strati di neuroni, e ad ogni neurone è associato un "jet".

Anche l'immagine da identificare viene rappresentata da uno strato neurale, ma di dimensioni più ampie; d'altronde l'immagine può presentare il viso in qualsiasi posizione.

Il processo di riconoscimento, detto "*dynamic link matching*" [24], consiste nel trovare le corrette corrispondenze/conessioni tra gli strati dei modelli dei volti disponibili e lo strato dell'immagine da identificare (ad esempio connettendo gli occhi sinistri dei modelli con quello sinistro dell'immagine). E' proprio questo procedimento a contrapporsi al normale funzionamento di una rete neurale di riconoscimento, in quanto le connessioni non sono fisse (durante il riconoscimento; in entrambe cambiano durante la fase di addestramento). Una volta individuato il corretto *mapping* tra modelli ed immagine, il riconoscimento può essere effettuato in maniera semplice attraverso il confronto dei "jet" appartenenti ai neuroni connessi.

Questa tecnica ha dimostrato invarianza alle traslazioni e robustezza alle deformazioni, come la rotazione in profondità (cioè dello sfondo). Se i "jet" fossero invarianti alle rotazioni si potrebbe ottenere anche invarianza alle rotazioni sul piano del volto, ma purtroppo, per come sono stati realizzati, non presentano questa caratteristica. Il principale problema di questo approccio rimane comunque la lentezza. [5][14]

2.3.4 Hidden Markov Model

Hidden Markov Model (HMM) si è dimostrato uno dei modelli più adatti, almeno dal punto di vista computazionale, allo scopo di riconoscere volti. HMM è nato con l'obiettivo di riconoscere le sequenze di proteine nel DNA, poi ha ottenuto ottimi risultati per problemi di OCR e di riconoscimento vocale.

I modelli di Markov sono uno strumento statistico in grado di riprodurre l'evoluzione di un sistema non deterministico mediante l'uso di stati.

Il numero di transizioni possibili da uno stato all'altro sono pari al quadrato del numero di stati. Le probabilità associate ad ogni transizione vengono memorizzate in una "matrice di transizione dello stato". I valori probabilistici possono essere calcolati osservando il comportamento storico del sistema; l'ampiezza di osservazione "all'indietro" determina l'ordine del modello (se dipendono dal solo stato precedente si parla di modello del I ordine). Oltre alla matrice descritta è necessario anche un vettore che descriva lo stato iniziale, o meglio le probabilità iniziali.

Gli HMM nascono nel caso in cui alcuni stati non siano direttamente osservabili (*hidden state*); quello che si cerca di fare è conoscere informazioni su di essi attraverso altri stati ad essi correlati in qualche modo. Il legame viene espresso attraverso una seconda matrice detta "matrice di confusione", i cui valori rappresentano le probabilità condizionate tra stati nascosti e stati direttamente osservabili.

I tratti somatici di una persona si presentano sempre nel medesimo ordine, per qualunque angolazione della foto. Il viso può quindi essere rappresentato da un modello bidimensionale HMM.

Dall'alto verso il basso si possono identificare cinque stati principali (superstati), riferiti a fronte, occhi, naso, bocca e mento, ognuno dei quali può essere suddiviso in più sottostati che esprimono ogni tratto somatico da sinistra verso destra.

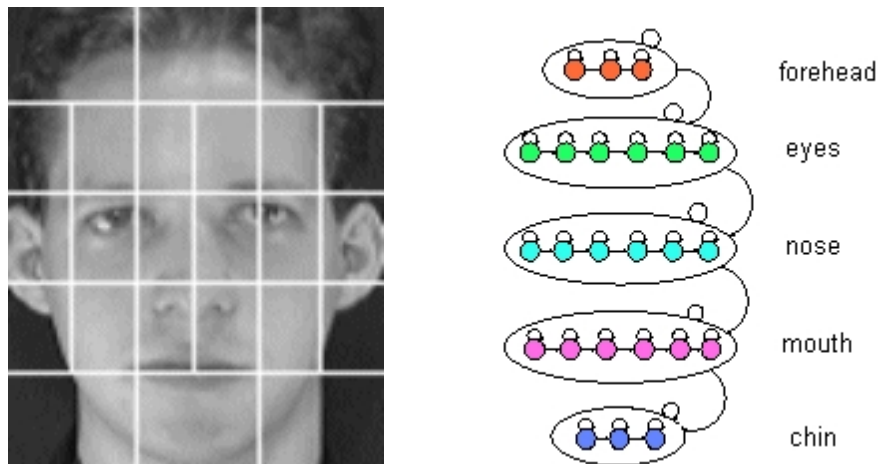


Figura 3.12: Suddivisione di un viso in stati

Il modello così ottenuto non è propriamente bidimensionale, ma ne è un'approssimazione, in quanto considerare stati verticali e stati orizzontali connessi uno ad uno sarebbe stato troppo oneroso computazionalmente. Un modello siffatto viene detto “*Embedded HMM*”.

E' stato rilevato sperimentalmente come gli *Embedded HMM* siano in grado di modellare dati bidimensionali in maniera nettamente migliore degli HMM monodimensionali, e con una complessità notevolmente inferiore a quelli bidimensionali.

Il modello è adatto a lavorare con qualsiasi oggetto che preservi le proprie caratteristiche in termini di posizione (dall'alto al basso e da destra a sinistra); è quindi adatta al riconoscimento facciale, in quanto le diversità tra i volti non consistono in variazione di posizione delle *feature* come occhi, naso e bocca.

L'utilizzo degli HMM per il riconoscimento facciale può essere sintetizzato in due fasi principali:

1. Addestramento del sistema: consiste nel determinare ed archiviare un modello per ogni individuo che si vuole riconoscere; di ogni individuo possono essere fornite più immagini, con espressioni e pose diverse.

2. Riconoscimento di un individuo: consiste nel creare un modello per l'individuo da identificare (in questo caso solitamente si può fare uso di una sola immagine) e nell'individuare tra i dati archiviati quello che maggiormente gli assomiglia.

La somiglianza viene espressa in termini probabilistici: maggiore è la probabilità maggiore è la somiglianza.

I problemi che si possono risolvere con gli HMM possono essere classificati in tre categorie:

- *Evaluation Problem*: sono problemi di *pattern matching*, che data una sequenza di osservazioni cercano di individuare l'HMM, tra quelli disponibili, che con maggiore probabilità possa averla generata. L'algoritmo utilizzato è di tipo *forward*. Questi problemi sono frequenti in programmi di riconoscimento vocale, dove ogni pronuncia definisce un diverso HMM.
- *Decoding Problem*: si tratta anche in questo caso di problemi di *pattern matching*. Dato un modello HMM si cerca di individuare la sequenza di stati nascosti che con maggiore probabilità possa aver generato una serie di osservazioni rilevate. Questi problemi sono frequenti sia in programmi di riconoscimento di pattern, sia in quelli di traduzione lessicale, dove il significato di una parola dipende dal contesto in cui si trova.
- *Training*: non si tratta di problemi di riconoscimento come i precedenti, quanto di problemi di apprendimento. Consistono nella generazione di un HMM che meglio realizzi la sequenza di osservazioni (*training set*). In termini matematici si tratta di massimizzare una funzione di verosimiglianza (*likelihood*).

Si stima che la probabilità di errore delle applicazioni che usano gli HMM sia intorno al 2-3%, anche se il comportamento di questo modello è quello di fornire in ogni modo un risultato: quello più probabile. Questo costituisce un limite, ed è dovuto proprio al fatto che il riconoscimento è stato modellato come processo probabilistico.

[15][16]

Capitolo 3

Riconoscimento del volto in presenza di alterazioni digitali delle immagini

In questo capitolo sarà introdotta la problematica legata alle alterazioni delle immagini nel riconoscimento del volto. Dopo una panoramica sulle alterazioni geometriche e sulla digital beautification, sarà approfondita la problematica legata al morphing.

3.1 Distorsioni geometriche e digital beautification

L'alterazione digitale delle immagini si può dividere in due categorie: *geometrica* e *appearance-based* (basate sull'apparenza). [26]

La prima categoria include trasformazioni che sono tipicamente introdotte sia dall'acquisizione che dalla stampa delle immagini. Un esempio possono essere le modifiche effettuate sull'aspect ratio delle immagini. Tipicamente questo tipo di alterazioni è involontario e dovuto al processo di acquisizione o di stampa.

In altri casi un soggetto potrebbe decidere di modificare deliberatamente la foto del volto, ad esempio con il semplice obiettivo di renderla più gradevole. Questo processo è noto come digital beautification e può facilmente essere realizzato con

strumenti online. Esistono infine casi, oggetto di questo lavoro di tesi, in cui le persone possono tentare di modificare l'immagine con scopi criminali. È questo il caso del morphing che sarà qui trattato in maniera più approfondita.

Due tipi di alterazioni che agiscono sulle proporzioni dell'immagine sono *vertical extension* e *vertical contraction*.

La contrazione verticale (Figura 4.1c) viene realizzata comprimendo verticalmente l'immagine mentre l'ampiezza viene mantenuta invariata.

Al contrario nell'estensione (Figura 4.1d) il valore dell'altezza viene aumentato lasciando sempre invariata l'ampiezza dell'immagine.

Un altro esempio di alterazione geometriche è la *barrel distortion*.

La *barrel distortion* (distorsione barile) (Figura 4.1b) è una delle più comuni distorsioni a lente esistenti e viene tipicamente introdotta da dispositivi a bassa qualità di acquisizione delle immagini. Viene così nominata per l'effetto "a barile" che produce sull'immagine, infatti le coordinate dei punti in questa alterazione vengono mappati sulla superficie di una sfera immaginaria.

Questo tipo di alterazioni nella maggior parte dei casi non sono intenzionali, poiché possono essere prodotte in modo inaspettato dalle tecnologie usate, come il risultato di un errore di stampa.

La seconda categoria di trasformazioni include tutte le alterazioni che sono normalmente effettuate tramite l'utilizzo di software per l'editing delle immagini.

Queste alterazioni sono introdotte per modificare l'aspetto dell'oggetto raffigurato nell'immagine; nella maggior parte dei casi per renderlo molto più attraente; un esempio possono essere l'editing di immagini per fini pubblicitari.

Negli ultimi anni questa categoria di trasformazioni viene molto utilizzata nei software di chirurgia plastica, grazie ai quali vengono mostrati ai pazienti i risultati prodotti da un ipotetico intervento di chirurgia estetica.

In [26] viene mostrato un esempio di *digital beautification* (abbellimento digitale) tramite l'uso di LiftMagic [27], un software di chirurgia estetica.

Si tratta di un'applicazione web che presenta un'interfaccia molto semplice e consente il caricamento di un'immagine, e la simulazione di differenti trattamenti di chirurgia plastica.

Nel software sono disponibili 17 trattamenti, 16 dei quali sono trattamenti locali (iniezioni per la fronte, aumento delle labbra, eccetera), e un trattamento che integra tutti quelli locali.

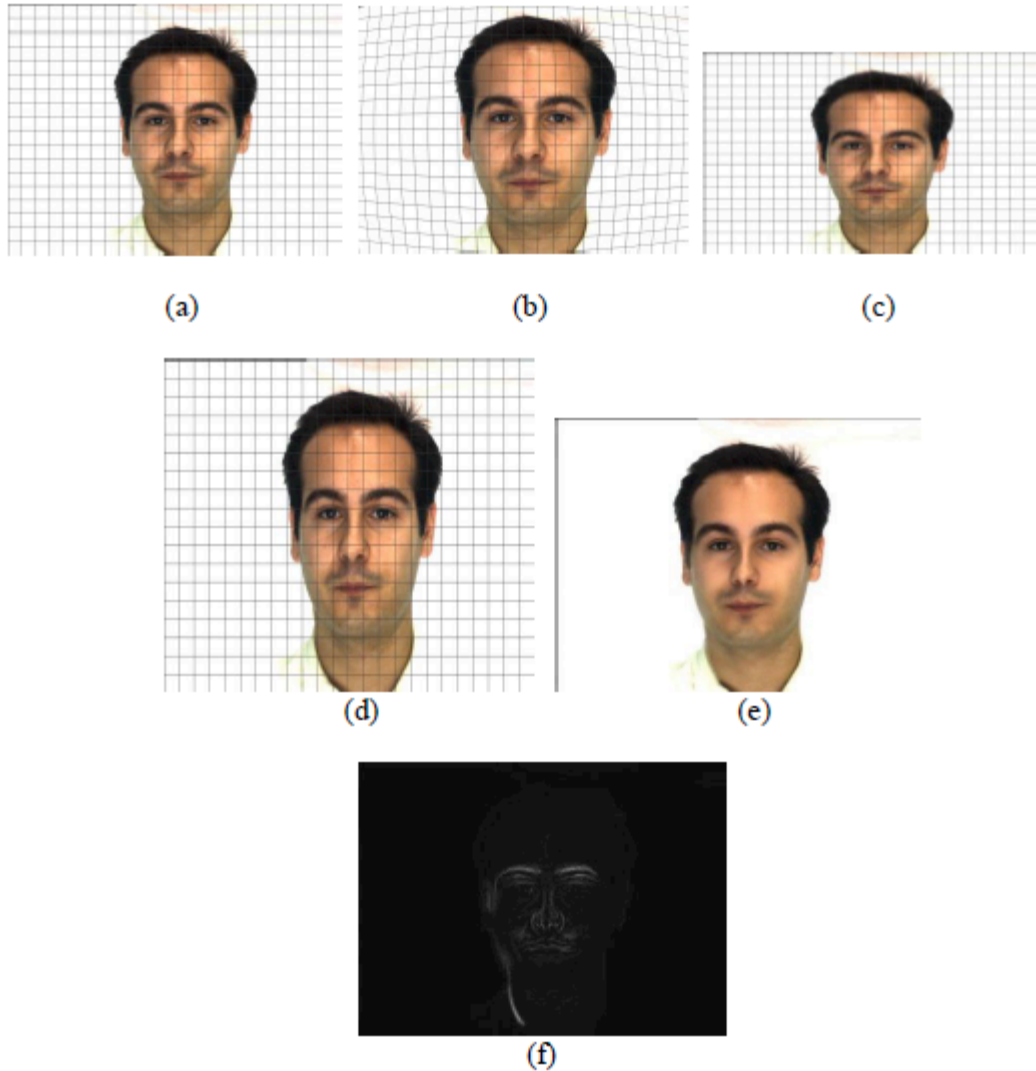


Figura 4.1: Esempi di alterazioni di immagini. (a) Immagine Originale. (b) *Barrel Distorsion*. (c) *Vertical Contraction*. (d) *Vertical Extension*. (e) *Digital Beautification*. (f) Differenza dei pixel tra l'immagine originale (a) e alterata tramite *digital beautification* (e)

Per ogni trattamento, una determinata barra di selezione permette di personalizzare il livello della modifica.

L'esempio di *digital beautification* in (Figura 4.1e) viene mostrato in figura.

Nei test svolti per questa tesi il *morphing* rientra nella categoria delle alterazioni *appearance-based*, in quanto viene utilizzato appunto per alterare i lineamenti e i tratti del viso di due persone.

3.2 Morphing

Obiettivo principale di questo lavoro di tesi è valutare la robustezza delle tecniche di riconoscimento del volto allo stato dell'arte rispetto a modifiche delle immagini operate tramite *morphing*. Come anticipato nell'introduzione tale tipo di alterazioni potrebbero essere operate con scopi criminali legati all'utilizzo di documenti di identità con elementi biometrici e, in particolare, del passaporto elettronico che utilizza il volto come caratteristica biometrica primaria per la verifica di identità.

Il lavoro di tesi ha richiesto quindi:

- La creazione di immagini *morphed* a partire da volti di diversi soggetti;
- L'esecuzione di test con software commerciali per il riconoscimento del volto al fine di valutarne la robustezza.

3.3 Morphing software

I software usati in questo lavoro di tesi per effettuare il *morphing* sono cinque software freeware che non impongono l'obbligo di creare un account per poter essere utilizzati.

In questa sezione saranno illustrate tutte le caratteristiche dei software esponendone le differenze.

3.3.1 Morphthing

Morphthing.com [18] (Figura 4.4) è un'applicazione web in cui i visitatori possono caricare le proprie foto o utilizzare quelle date a disposizione dal sito ed effettuare il *face morphing* sulle immagini desiderate.



Figura 4.2: Home Page di Morphthing

L'utilizzo del software è molto semplice grazie al sito che guida l'utente nei diversi passaggi.

Per effettuare il *morphing* sulle foto caricate bisogna prima prepararle, cioè bisogna posizionare sull'immagine quattro punti verdi, i *landmarks* (Figura 4.5), che indicano le diverse parti del viso, una volta posizionati un'anteprima mostrerà il risultato della selezione; se è soddisfacente si può continuare con il *morphing*, altrimenti è possibile passare alla modalità avanzata cliccando su "Try Again".

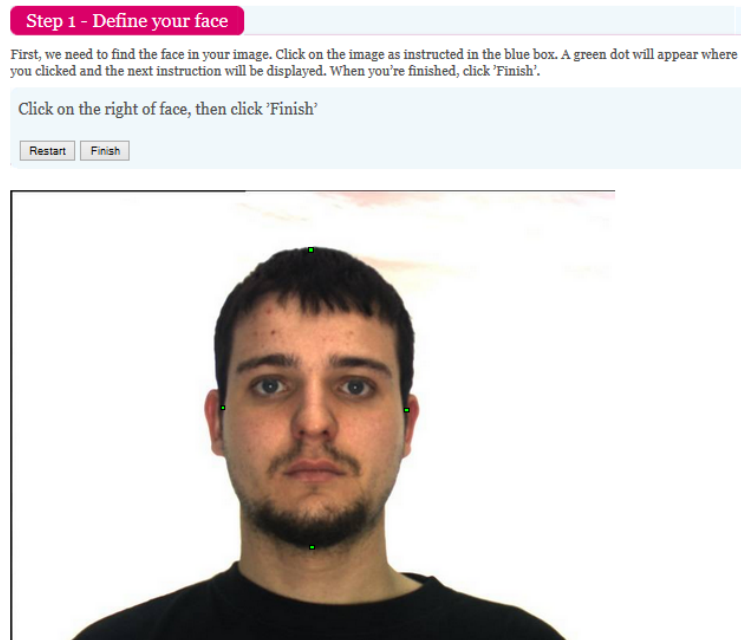


Figura 4.3: Esempio landmarks di MorphThing

Nella modalità avanzata invece di utilizzare solo quattro *landmarks* il sito guida l'utente al posizionamento di ulteriori punti per specificare i lineamenti del volto: sopracciglia, occhi, naso, labbra, fronte e mandibola.

Una volta posizionati il sito richiede di selezionare l'etnia dei soggetti presenti nelle foto, successivamente è sufficiente cliccare il pulsante "Morph" per fare iniziare il processo di trasformazione delle immagini al software. Se si utilizzano foto già presenti sul sito web si salta il processo di preparazione.

Le foto caricate possono essere classificate dall'utente che le ha caricate come immagini private o pubbliche, cioè utilizzabili da tutti.

Sul sito, oltre al normale *morphing*, è possibile effettuare il "Morph Baby" in cui il risultato sarà il possibile volto di un bambino che assomiglia ai soggetti delle due foto utilizzate.

L'applicazione accetta immagini sia in formato PNG che JPEG, l'immagine creata dal software è in formato JPEG di dimensioni 300 x 400 con una risoluzione di 72dpi.

Questa applicazione web è uno dei software più utilizzati per il *morphing* nel mondo, solo dopo sei mesi dalla fondazione del sito sono stati creati un milione di *morphing*.

3.3.2 Fotomorph

Fotomorph è un editor fotografico con funzioni per la creazione di effetti di *morphing* tra due immagini.

Il risultato finale di *morphing* di questo programma è un'animazione costituita da un numero di immagini che è possibile scegliere; al termine del lavoro l'utente ha la possibilità di esportare l'intera animazione e anche i singoli frame che la compongono.

Il layout del software è semplice e intuitivo, una volta installato il programma contiene già dei *morphing* per fare sì che i nuovi utenti abbiano a disposizione degli esempi da cui prendere spunto.

L'editor è composto da quattro finestre principali.

- La prima finestra è *Projects*, in cui si possono creare nuovi progetti e gestire quelli già esistenti, e manipolare le sequenze, la velocità e la risoluzione delle animazioni create dal *morphing* di due foto.
- Segue la finestra *Images* (Figura 4.6) che permette di caricare, modificare, e allineare le immagini che si desidera utilizzare in modo tale che i lineamenti del volto siano sovrapposti per la prossima finestra *Control*.
- Nella finestra *Control* (Figura 4.7) risiede il passaggio fondamentale per il *morphing*: il posizionamento dei *landmarks*; in questa finestra è disponibile un'anteprima del *morphing* per mostrare il risultato finale durante il posizionamento dei punti.
- Nella quarta finestra *Animation* il programma mostra l'animazione finale del *morphing*.

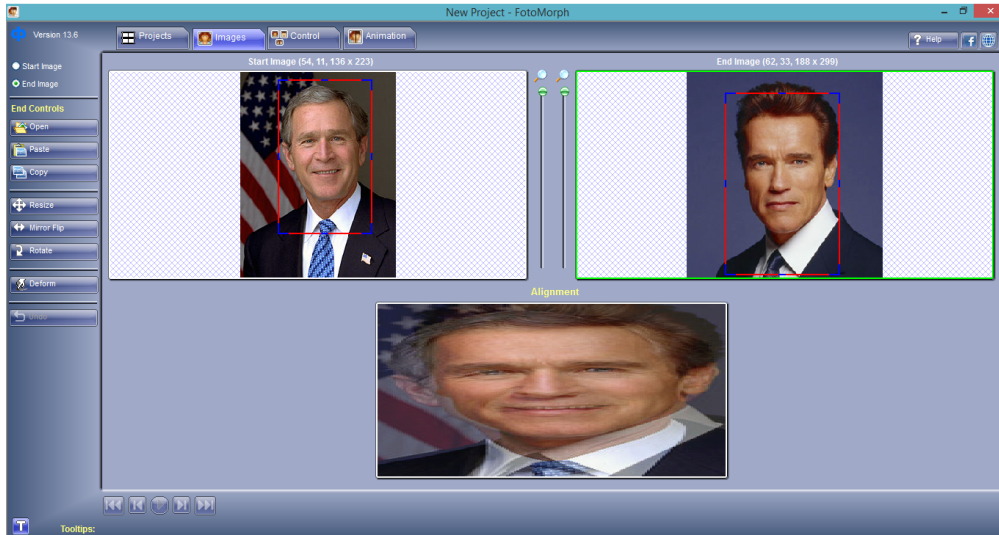


Figura 4.4: Finestra Images di Fotomorph

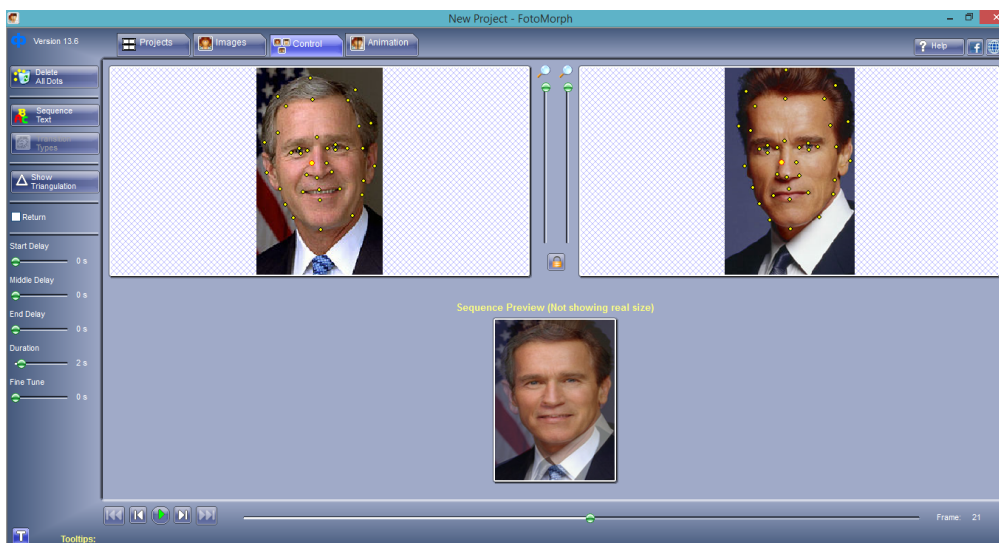


Figura 4.5: Finestra Control di Fotomorph

In Fotomorph sono supportati diversi formati grafici: BMP, JPEG, PNG, TIFF e GIF e i lavori sono esportabili in GIF, SWF, HTML, AVI, screensaver o come file EXE. Le immagini esportate da questo software hanno una dimensione di 300 x 400 con una risoluzione di 96dpi.

3.3.3 Sqirlz Morph

Sqirlz Morph [19] (Figura 4.8) come Fotomorph è un editor di immagini in cui il *morphing* finale è un'animazione di una sequenza di immagine in cui è possibile sceglierne il numero.

Poche risultano essere le differenze con il software precedente: la selezione dei pulsanti per la creazione o rimozione dei nuovi punti e il movimento dei punti creati invece che con la normale gestione del mouse, e la dimensione e risoluzione delle immagini esportate, rispettivamente 768 x 576 e 96pi.

Sqirlz Morph accetta il caricamento soltanto immagini in formato JPEG, mentre può esportare l'animazione in formato AVI, immagini in JPEG o l'intero lavoro in GIF e BITMAP.



Figura 4.6: Finestra di Sqirlz Morph

3.3.4 Face Morpher Lite

Face Morpher Lite [20] è la versione gratuita del software Face Morpher della compagnia Luxand [23] che si occupa di intelligenza artificiale e tecnologie per il riconoscimento biometrico.

La differenza fra i due programmi oltre al layout ridotto all'essenziale della versione gratuita è la possibilità di creare il *morphing* fra più immagini nella versione a pagamento.

Diversamente dai software precedenti una volta caricate le immagini Face Morpher Lite riesce a identificare autonomamente le diverse parti del viso e propone all'utente un risultato immediato.

Se non soddisfatto l'utilizzatore può accedere alla modalità di ritocco manuale in cui si può modificare la posizione preimpostata dal programma dei *landmarks* sulle immagini (Figura 4.9).



Figura 4.7: Ritocco manuale in FaceMorpherLite

L'utilizzo è semplificato dalla guida, attivata all'inserimento dell'immagine, che mostra i passi da seguire per realizzare il *morphing* (Figura 4.10).



Figura 4.8: Guida di FaceMorpherLite

Face Morpher Lite richiede il caricamento di immagini in formato JPEG, mentre come formato di uscita è possibile salvare l'animazione in AVI, creare una GIF animata, o salvare solo il frame corrente in JPEG.

Le dimensioni delle immagini ottenute dal *morphing* sono 256 x 256 con una risoluzione di 96dpi.

3.3.5 Face On Face

Faceonface.com [21] (Figura 4.11) è un'applicazione web molto simile a Morphthing.com.

Anche in questo sito è possibile caricare foto o utilizzare quelle già esistenti, effettuare il *morphing* senza essere registrati, settare le proprie immagini come pubbliche o come private.

A differenza di Morphthing.com faceonface.com richiede il caricamento di file solo JPEG che è lo stesso formato di uscita.

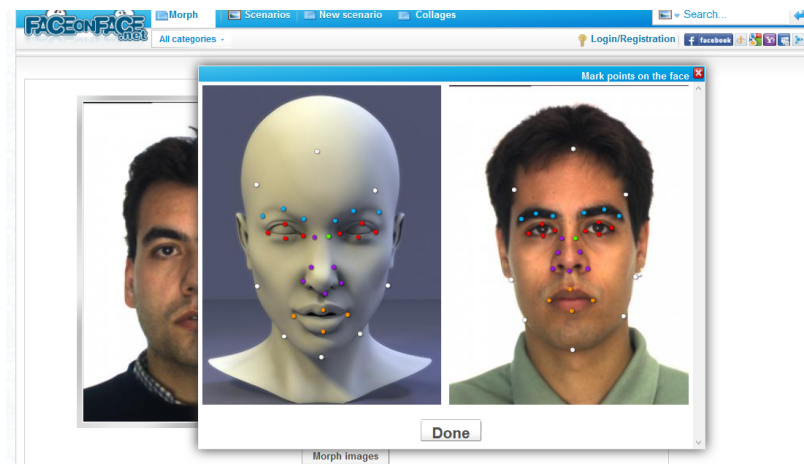


Figura 4.9: Finestra del posizionamento dei landmarks in Faceonface

Anche questo sito dà la possibilità all'utente di utilizzare il Baby Morphing, e altre due ulteriori funzioni: Aging per modificare una foto in modo da rendere il soggetto più vecchio, e Cartoon per creare il cartone animato della foto caricata.

Le dimensioni e le risoluzioni delle immagini *morphing* sono le stesse di Fotomorph.

3.3.6 Confronto tra i Software di Morphing

La tabella riprodotta in Figura 4.12 mostra un riassunto delle maggiori differenze per quanto riguarda la risoluzione, il formato e il numero massimo di *landmarks* di ciascun software.

Per mostrare la differenza dei risultati fra i diversi programmi, prendendo come riferimento la coppia raffigurata in Figura 4.13, vengono mostrate le immagini prodotte da ogni software (Figura 4.14).

Morphing Software / Caratteristiche	Risoluzione	Formato input	Formato output	Numero massimo landmarks
Morphthing	300x400 (72 dpi)	PNG, JPEG	JPEG	36
FotoMorph	300x400 (96 dpi)	BMP, JPEG, PNG, TIFF, GIF	GIF, SWF, HTML, AVI, EXE	infiniti
Sqirlz Morph	768x576 (96 dpi)	JPEG	AVI, JPEG, GIF, BITMAP	infiniti
Face Morpher Lite	256x256 (96 dpi)	JPEG	JPEG, AVI, GIF	40
Face on Face	300x400 (96 dpi)	JPEG	JPEG	33

Figura 4.10: Tabella di confronto tra i cinque software di morphing



Figura 4.11: Immagini di partenza per la realizzazione del morphing



Figura 4.12: Risultati di morphing da in alto a sinistra a destra di: Morphthing, FotoMorph, FaceonFace, Sqirlz Morph, Face Morpher Lite

Dalle immagini è possibile riconoscere le diverse risoluzioni dei software, soprattutto nei dettagli come lineamenti del viso, segni particolari come nevi della pelle, o attaccatura dei capelli presenti in maniera poco definita nel *morphing* prodotto da Face Morpher Lite, e invece, al contrario, chiari e distinti nel risultato realizzato da Sqirlz Morph e Morphthing.com.

A parte questi particolari, in generale i volti risultanti dai morphing dei software sono riconducibili tutti alla stessa ipotetica persona, essendo tutti molto simili tra loro.

3.4 Test con software commerciali

I test effettuati per valutare la robustezza dei sistemi di riconoscimento si basano su una serie di prove di riconoscimento di tipo *genuine* e *impostor* (autentico / impostore). In un test *genuine* sono confrontate due immagini di volti dello stesso individuo, mentre nel test *impostor* vengono confrontate due immagini di persone diverse. In ogni test *genuine* / *impostor* la prima immagine si suppone sia stata acquisita nel corso di una fase di “rilascio” del documento (quella inclusa nel documento) e la seconda in una fase di “verifica”.

I test consistono nell’identificazione di due soggetti in un’immagine alterata digitalmente tramite *morphing*, realizzata con le due immagini raffiguranti i soggetti stessi.

Su un database di volti di persone sono state scelte le coppie che più tra loro si somigliavano per realizzare i *morphing* che sarebbero stati testati con i programmi di riconoscimento.

Queste coppie sono state prelevate dal *AR Face Database* [25], un database che contiene 4000 immagini frontali di 126 soggetti rilevate sotto differenti condizioni in due sessioni diverse a due settimane di distanza, e che rispettano i requisiti standard per un documento eMRTD.

All’interno del database per ogni persona sono disponibili diverse immagini, ognuna con pose e illuminazione differenti.

In questi test sono state utilizzate nei *morphing* e nei riconoscimenti le pose 1 e 14 (Figura 4.2) in quanto presentano un’espressione neutrale e una buona illuminazione, in modo da agevolare sia la fase di *morphing* che quella di riconoscimento.



Figura 4.13: Immagini selezionate dal database AR. A sinistra la posa 1 e a destra la posa 14

Per ogni soggetto una delle due immagini è stata usata per il *morphing* mentre l'altra viene utilizzata per il test di riconoscimento in modo da simulare in maniera corretta le fasi di "rilascio" e "verifica".

I test sono stati effettuati utilizzando cinque software di *morphing* commerciale freeware e due software anche questi disponibili in commercio per il riconoscimento facciale.

Nelle immagini prodotte dal *morphing*, indipendentemente dalla qualità del software utilizzato, sono sempre presenti delle ombre e piccoli difetti dovuti all'incongruenza dei contorni e lineamenti del viso di due soggetti.

Per riuscire a dare un'impressione reale del soggetto presente nell'immagine *morphed*, i risultati dei *morphing* venivano ritoccati e perfezionati con un editor di immagini open source: GIMP (GNU Image Manipulation Program) (Figura 4.3).

[17]



Figura 4.14: Esempio di immagine morphed non ritoccata a sinistra, e ritoccata a destra

Per simulare una realistica situazione di controllo di un sistema ABC le soglie di accettazione per il riconoscimento dei software sono state predisposte secondo gli stessi criteri stabiliti dalla FRONTEX (the European Agency for the External Borders of the Member States of the European Union), i quali definiscono che il livello massimo del FAR (*False Accept Rate*), pari a 0.1% con un valore di FRR (*False Rejection Rate*) del 5%. Naturalmente il valore del FAR varia in base alla soglia di similarità utilizzata; per i test sono state adottate le soglie suggerite dai produttori dei software per soddisfare il requisito di un FAR 0.1%.

3.4.1 VeriLook 5.3

VeriLook 5.3/MegaMatcher 4.4 è un software per il riconoscimento facciale sviluppato da Neurotechnology [22] che si occupa della creazione di sistemi e algoritmi per il riconoscimento biometrico (Figura 4.15).

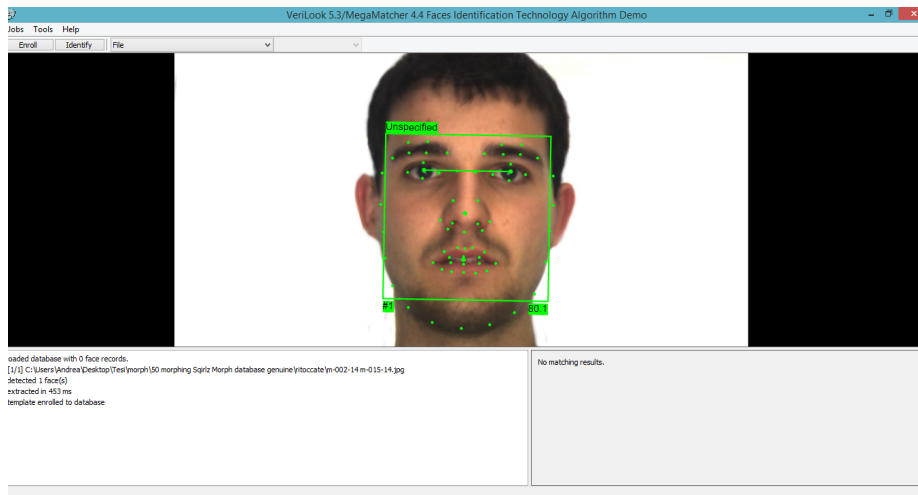


Figura 4.15: Face detection di VeriLook 5.3/MegaMatcher 4.4

Implementato su algoritmi sviluppati dalla Neurotechnology questo software dispone di numerose funzioni per il riconoscimento:

- possibilità di effettuare sia confronti 1-a-1 che 1-a-molti
- utilizzo della fotocamera del pc per il riconoscimento con una immagine
- localizzazione del volto istantaneamente con l'utilizzo della fotocamera
- effettuare *matching* di volti da video
- realizzazione del riconoscimento in meno di un secondo.
- possibilità di salvare le *feature* di un volto sul database del programma
- estrazione, analisi e combinazione delle *feature* di un volto da più immagini
- confrontare le *feature* di una immagine con quelle presenti nel database

Nella finestra delle opzioni è possibile configurare i vari parametri per la *face detection*, estrazione delle *feature*, il caricamento delle immagini, e l'identificazione; per esempio è possibile imporre al programma delle soglie minime di tolleranza per l'accettazione dei volti, come la distanza degli occhi per l'azione di *face detection*.

Durante la fase dei test la soglia di accettazione sulla risoluzione delle immagini è stata settata al minimo per far sì che tutti i *morphing* non venissero scartati per la loro qualità.

VeriLook 5.3 è disponibile sia per piattaforme Windows, Linux o Mac OS X.

3.4.2 Luxand SDK Face Matcher

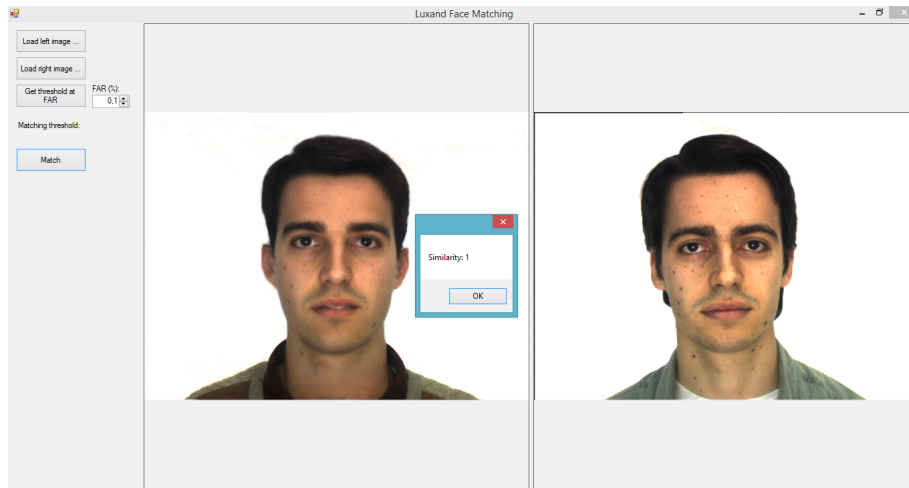


Figura 4.16: Matching con Luxand SDK Face Matcher

Luxand FaceSdk [23] è una libreria cross-platform che implementa funzioni per la localizzazione del volto, l'estrazione dei *feature* facciali, e identificazione di volti su immagini e video (Figura 4.16).

Si tratta di una libreria dinamica disponibile in 32 o 64 bit per Windows e Linux, 64-bit MacOS X, iOS, Android. The SDK contiene i file di intestazione interfaccia e esempi di applicazioni per C++, Microsoft Visual C++ 6.0/2005/2008, Visual Basic .NET 2005/2008, Microsoft C# .NET 2005/2008, Borland Delphi 6.0/7.0, Netbeans (Java), Xcode 4.2+ (iOS), Eclipse ADT (Android), Visual Basic 6.0 and C++Builder 6.0.

Questa SDK è dotata di Tracker API per il riconoscimento e monitoraggio nei video dei volti in tempo reale e ne registra tutte le informazioni biometriche che è in grado di rilevare nel video.

Nel software l'unico parametro modificabile è il FAR.

Come nel programma precedente anche con questa libreria è possibile controllare la telecamera del pc, e il riconoscimento viene effettuato in meno di un secondo.

3.5 Risultati

In questo paragrafo saranno mostrati i risultati ottenuti dal confronto dei *morphing* con i relativi soggetti con cui sono stati realizzati.

Nella prima parte di test per ognuno dei cinque software sono stati effettuati i *morphing* di 19 coppie di persone (uomini e donne) che presentano lineamenti e caratteristiche del volto simili.

Per i due software di riconoscimento i valori risultanti e le soglie di identificazione sono differenti; per quanto riguarda Verilook 5.3/MegaMatcher 4.4 se il risultato del *matching* è superiore o uguale a 36 il soggetto nelle immagini confrontate viene riconosciuto, mentre in Luxand Face SDK il valore di identificazione deve essere maggiore o uguale di 0.999.

In seguito sono presentati i risultati del *matching* dei *morphing* (Figura 4.17, 4.18, 4.19, 4.20, 4.21).

I risultati sono stati marcati con tre colori differenti: i valori in blu sono quelli in cui entrambi i soggetti di un *morphing* sono stati riconosciuti nel *matching*, i risultati in verde sono quelli in cui il soggetto è stato riconosciuto ma l'altro ha avuto un esito negativo, e quelli in rosso sono i soggetti che non sono stati riconosciuti.

Morphthing

N° Test Morphing	VeriLook 5.3/Megamatcher 4.4 Algorithm Demo		Luxand Face Matching	
	Soggetto n°1	Soggetto n°2	Soggetto n°1	Soggetto n°2
1	180	144	0.99999	0.99988
2	184	180	0.99996	1
3	226	100	1	0.96344
4	252	141	0.99999	0.99999
5	180	171	0.99999	0.99998
6	296	178	1	0.99998
7	333	138	1	0.99994
8	150	183	0.99881	0.99799
9	247	119	1	0.99572
10	139	183	0.99999	0.99788
11	160	207	0.99999	0.99948
12	276	193	0.99997	0.99996
13	129	135	0.99916	0.99999
14	249	170	0.99999	0.99799
15	153	224	0.99966	0.99999
16	183	124	0.98712	0.99593
17	219	141	0.99991	0.99968
18	160	142	0.99948	0.94249
19	163	270	0.99892	0.99865

Figura 4.17: Risultati dei matching tra i morphing del software Morphthing

Fotomorph

N° Test Morphing	VeriLook 5.3/Megamatcher 4.4 Algorithm Demo		Luxand Face Matching	
	Soggetto n°1	Soggetto n°2	Soggetto n°1	Soggetto n°2
1	163	193	1	0.99949
2	142	164	0.99749	0.99998
3	329	71	1	0.93989
4	182	101	0.99989	0.99966
5	134	165	0.99981	0.99999
6	191	193	1	0.99841
7	253	158	0.99994	0.99896
8	224	157	0.99953	0.99942
9	310	139	0.99999	0.99458
10	79	117	0.99999	0.99919
11	221	142	1	0.99154
12	171	270	1	0.99994
13	156	126	0.99999	0.99999
14	208	161	0.99027	0.99278
15	113	131	0.99999	0.99994
16	119	154	0.99199	0.99958
17	217	136	0.99999	0.99967
18	196	131	0.99993	0.99872
19	200	145	0.99397	0.99004

Figura 4.18: Risultati dei matching tra i morphing del software Fotomorph

Sqirlz Morph

N° Test Morphing	VeriLook 5.3/Megamatcher 4.4 Algorithm Demo		Luxand Face Matching	
	Soggetto n°1	Soggetto n°2	Soggetto n°1	Soggetto n°2
1	207	219	0.99999	0.99998
2	207	233	0.99941	1
3	357	100	1	0.99408
4	292	134	1	0.99984
5	163	212	0.99943	1
6	286	342	0.99999	0.99976
7	378	224	0.99999	0.99973
8	249	211	0.99531	0.99999
9	221	173	0.99999	0.97698
10	104	221	0.99999	0.99983
11	150	220	1	0.99988
12	245	311	0.99999	0.99999
13	100	215	0.99999	1
14	264	231	0.99994	0.99996
15	153	219	0.99999	0.99999
16	170	192	0.99629	0.99996
17	222	222	0.99971	0.99996
18	209	219	0.99693	0.99919
19	177	292	0.99118	0.99809

Figura 4.19: Risultati dei matching tra i morphing del software Sqirl Morph

Face Morpher Lite

N° Test Morphing	VeriLook 5.3/Megamatcher 4.4 Algorithm Demo		Luxand Face Matching	
	Soggetto n°1	Soggetto n°2	Soggetto n°1	Soggetto n°2
1	128	138	0.99963	0.99992
2	88	134	0.99995	0.99998
3	175	76	0.99999	0.99961
4	230	89	0.99990	0.99905
5	71	85	0.99225	0.99972
6	150	176	0.99999	0.99999
7	153	109	0.99985	0.99687
8	121	207	0.95715	0.97256
9	168	85	0.99419	0.96561
10	108	107	0.99931	0.99731
11	71	146	0.99999	0.99745
12	173	154	0.99999	0.99995
13	82	145	0.99876	0.99999
14	142	118	0.99749	0.99553
15	68	145	0.99955	0.99997
16	85	72	0.83946	0.98704
17	118	153	0.99960	0.99991
18	135	109	0.99987	0.94022
19	136	190	0.97999	0.99742

Figura 4.20: Risultati dei matching tra i morphing del software Face Morpher Lite

Face on Face

N° Test Morphing	VeriLook 5.3/Megamatcher 4.4 Algorithm Demo		Luxand Face Matching	
	Soggetto n°1	Soggetto n°2	Soggetto n°1	Soggetto n°2
1	183	215	0.99999	0.99992
2	171	126	0.99957	1
3	362	116	1	0.96731
4	249	138	0.99999	0.99962
5	210	142	0.99974	0.99998
6	228	213	0.99999	0.99953
7	410	119	0.99999	0.99987
8	157	142	0.90717	0.99932
9	229	120	0.99999	0.98278
10	147	197	0.99999	0.99833
11	157	143	0.99999	0.99998
12	296	249	0.99999	0.99982
13	112	176	0.99999	1
14	244	199	0.99999	0.99859
15	113	249	0.99976	0.99999
16	175	140	0.99462	0.99695
17	165	173	0.99964	0.99999
18	160	160	0.99996	0.99898
19	225	284	0.99689	0.99988

Figura 4.21: Risultati dei matching tra i morphing del software Face on Face

Come è possibile vedere dai risultati ogni soggetto è stato riconosciuto in ogni *morphing* dal software VeriLook 5.3/MegaMatcher 4.4 con il FAR suggerito da FRONTEX.

Se fossero state impostate delle soglie per la risoluzione delle immagini i programmi con una bassa qualità delle immagini come Face Morpher Lite non sarebbero riusciti a superare positivamente il confronto.

Al contrario in Luxand Face Matching non tutti gli esiti sono stati positivi; ecco la percentuale di successo dei software di *morphing* contro il sistema di riconoscimento:

- Morphthing: 58%
- Fotomorph: 47%
- Sqirlz Morph: 68%
- Face Morpher Lite: 42%
- Face on Face: 58%

I risultati di questo software si trovano tutti in un range tra il 40% e il 70%.

Questo significa che la probabilità per un documento falsificato con un'immagine *morphed* di superare un sistema di controllo basato sul riconoscimento del volto è di

almeno due volte su tre per quanto riguarda Sqirlz Morph, e una volta su due per gli altri software.

I risultati tra i due software sono molto diversi, ciò è dovuto ai differenti algoritmi di riconoscimento ed estrazione delle *feature* utilizzati.

Questo ci mostra come le situazioni possono variare a seconda del tipo di sistema usato nel riconoscimento.

In seguito per ogni software saranno mostrati due esempi di *morphing* corrispondenti ai risultati ottenuti, e le relative coppie, una composta da uomini e l'altra da donne.

Le prime due immagini sono quelle usate nel *morphing*, mentre la terza è l'immagine realizzata dal software.

Morphthing



Figura 4.22: Morphing N°2 Morphthing



Figura 4.23: Morphing N°10 Morphthing

Fotomorph

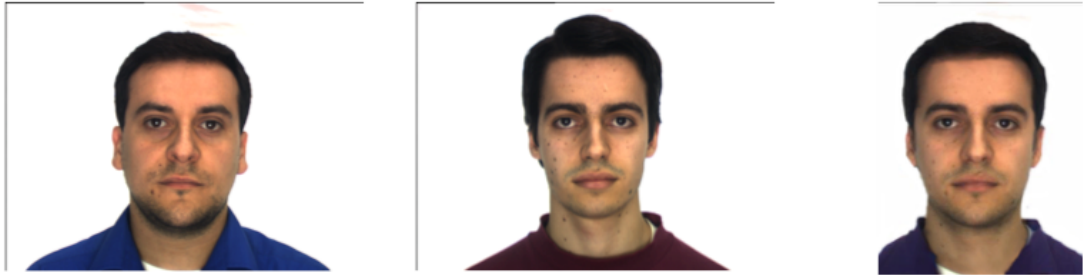


Figura 4.24: Morphing N°6 Fotomorph

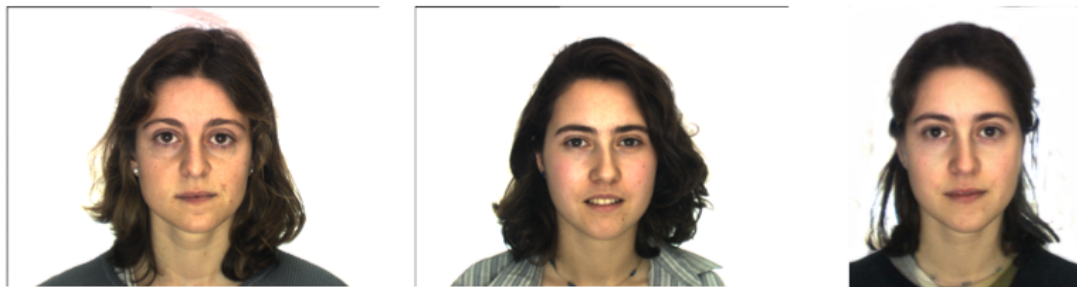


Figura 4.25: Morphing N° 9 Fotomorph

Sqirlz Morph

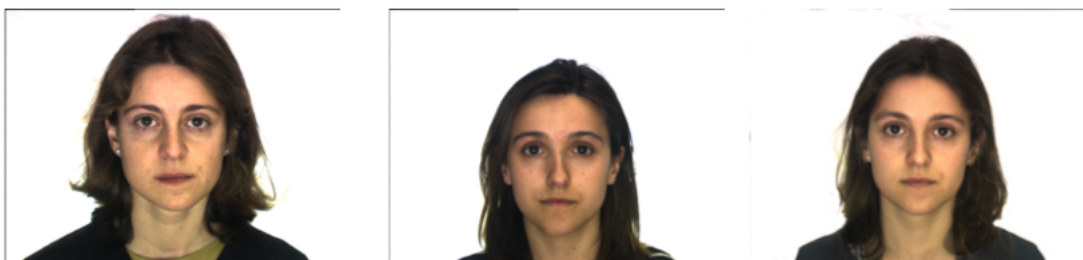


Figura 4.26: Morphing N°7 Sqirlz Morph



Figura 4.27: Morphing N°15 Sqirlz Morph

Face Morpher Lite

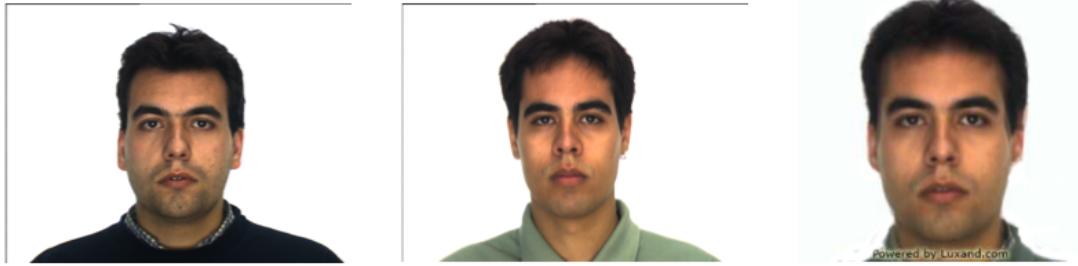


Figura 4.28: Morphing N°4 Face Morpher Lite

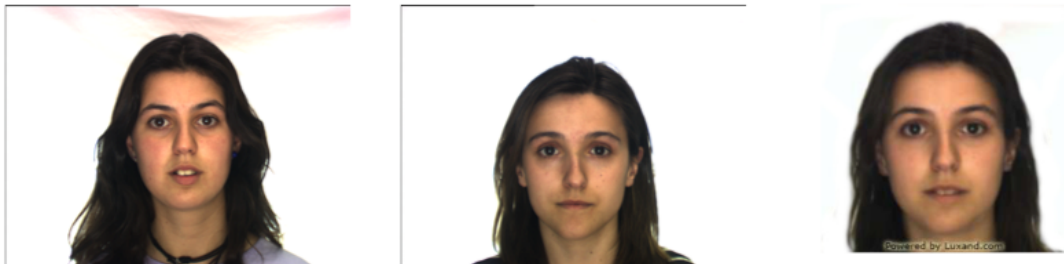


Figura 4.29: Morphing N°18 Face Morpher lite

Face on Face

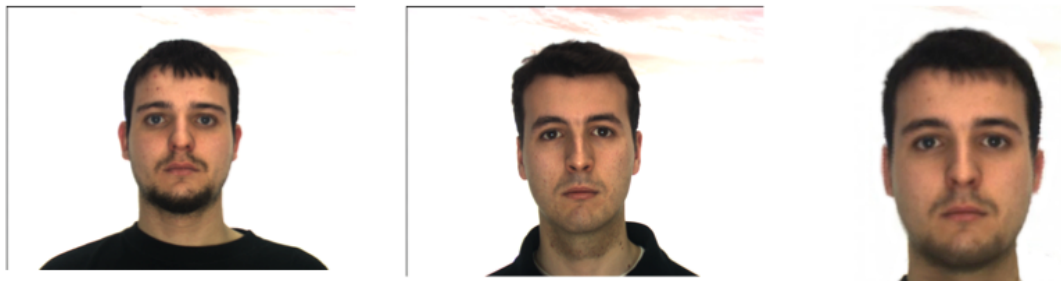


Figura 4.30: Morphing N°1 Face on Face

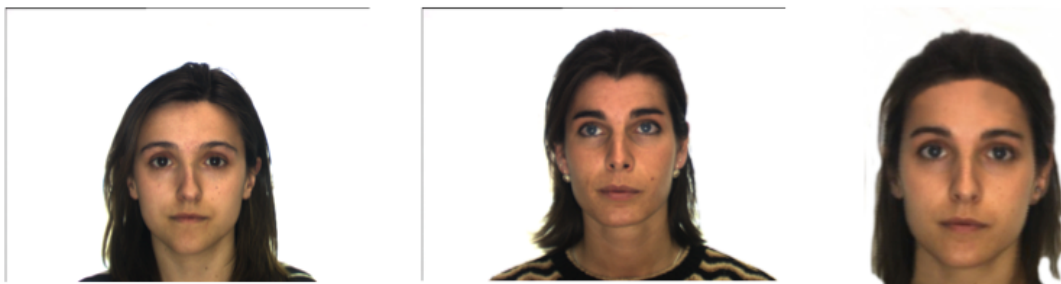


Figura 4.31: Morphing N°8 Face on Face

Per approfondire la ricerca nella seconda parte di test sono stati svolti altri *morphing* con i due software che producono un'immagine *morphed* con maggiore risoluzione rispetto agli altri e hanno avuto esiti migliori nei *matching*, per simulare l'alta qualità delle foto utilizzate in un passaporto: Sqirlz Morph e Morphthing.

I *matching* dei *morphing* sono stati eseguiti con il programma della Luxand essendo più affidabile e preciso, e quindi avendo maggiori somiglianze con un sistema di controllo ABC.

Questi test sono stati realizzati su 100 coppie prelevate sempre dal *AR Face Database*, per ogni software di *morphing* sono state utilizzate 50 persone diverse.

Le iniziali dei nomi dei *morphing* indicano “m” se la coppia è composta due uomini, e “w” se invece sono due donne.

In Figura 4.22a e 4.22b sono mostrati i risultati:

Morphthng Morphing	Score		Sqrlz Morph Morphing	Score	
	Soggetto 1	Soggetto 2		Soggetto 1	Soggetto 2
m-001-14 m-031-14	0.99998	0.99998	m-001-1 m-031-1	0.99998	1
m-002-14 m-065-14	0.99720	0.99999	m-001-14 m-002-14	0.99996	0.99967
m-002-14 m-066-14	0.99917	0.99985	m-001-14 m-027-14	0.99999	0.99955
m-004-14 m-025-14	0.99408	0.99983	m-002-14 m-004-14	0.99999	0.99979
m-007-1 m-067-1	0.99743	0.99999	m-002-14 m-006-14	0.99932	0.99785
m-007-14 m-016-14	0.91747	0.67392	m-002-14 m-015-14	0.99999	0.999999
m-009-14 m-052-14	0.99990	0.99910	m-003-1 m-032-1	0.99798	0.999999
m-010-1 m-025-1	0.99971	0.99998	m-004-1 m-037-1	0.99999	0.99963
m-012-14 m-052-14	0.99999	0.99993	m-004-14 m-012-14	0.99998	0.99947
m-013-14 m-014-14	0.99932	0.99935	m-004-14 m-067-14	0.99999	1
m-013-14 m-042-14	0.99821	0.99999	m-007-1 m-065-1	0.99722	0.98719
m-014-14 m-042-14	0.99957	1	m-007-14 m-018-14	0.99253	0.99989
m-017-14 m-071-14	0.99997	0.99941	m-007-14 m-072-14	0.99987	0.99999
m-020-1 m-067-1	0.92682	0.99988	m-009-14 m-010-14	0.99906	0.99966
m-021-14 m-42-14	0.99740	0.99999	m-009-14 m-013-14	0.99567	0.99273
m-022-1 m-052-1	0.99404	0.99999	m-010-14 m-030-14	0.99983	0.99999
m-042-14 m-053-14	1	0.99999	m-012-1 m-054-1	0.99999	0.99999
m-052-1 m-067-1	0.99999	1	m-012-1 m-056-1	0.99999	0.99987
m-052-1 m-069-1	0.99995	0.99992	m-013-14 m-052-14	0.99925	0.99988
w-003-14 w-007-14	0.99999	0.99981	m-014-1 m-015-1	0.99996	0.99969
w-007-14 w-055-14	0.99941	0.99752	m-014-14 m-015-14	0.99984	1
w-008-14 w-014-14	0.99343	0.92039	m-015-1 m-042-1	0.99999	1
w-011-1 w-017-1	0.99641	0.99999	m-015-1 m-067-1	0.99999	0.99918
w-016-14 w-043-14	0.99999	0.99910	m-015-14 m-053-14	1	1
w-037-14 w-040-14	0.99998	0.99095	m-018-1 m-053-1	0.99999	0.99987
m-001-1 m-002-1	0.99999	0.99159	m-018-1 m-069-1	0.99999	0.99999
m-001-14 m-025-14	0.99979	0.99999	m-018-14 m-021-14	1	0.99999
m-002-1 m-004-1	0.99987	0.99691	m-018-14 m-074-14	0.99999	0.98529
m-002-1 m-015-1	0.99999	1	m-018-14 m-052-14	1	0.99999
m-002-1 m-022-1	0.99999	0.99489	m-020-14 m-031-14	0.99659	0.99999
m-002-14 m-009-14	1	0.99254	m-023-1 m-025-1	0.99993	0.99999
m-004-14 m-018-14	0.99999	0.99999	m-025-14 m-031-14	0.99996	0.99996
m-006-14 m-069-14	1	0.99914	m-025-14 m-065-14	0.99970	0.99989
m-007-1 m-016-1	0.92406	0.96861	m-027-1 m-030-1	0.99864	1
m-009-14 m-025-14	0.99955	0.99999	m-027-14 m-031-14	0.99985	0.99776
m-010-14 m-027-14	0.99978	0.99997	m-031-14 m-060-14	1	0.99885
m-013-14 m-018-14	0.99995	0.99999	m-032-1 m-067-1	1	0.99931
m-014-1 m-042-1	0.99987	0.99946	m-037-14 m-074-14	0.99833	0.93651
m-015-1 m-021-1	1	0.99999	m-053-1 m-022-1	0.99846	0.99980
m-015-1 m-052-1	0.99999	0.99997	m-065-1 m-067-1	1	0.99708
m-015-14 m-042-14	0.99999	0.99999	w-003-1 w-037-1	0.99938	0.99978
m-018-1 m-056-1	0.99999	0.99229	w-007-1 w-057-1	0.99998	0.98625
m-021-14 m-025-14	0.99999	0.99994	w-008-14 w-011-14	0.99531	0.99169
m-042-1 m-052-1	1	1	w-009-1 w-036-1	0.99335	0.95022
m-053-1 m-067-1	0.99999	0.99982	w-016-14 w-038-14	0.99971	0.99989
m-054-14 m-036-14	0.99978	0.99307	w-016-14 w-045-14	0.99949	0.99735
w-008-14 w-038-14	0.89575	0.99998	w-017-1 w-036-1	1	0.99984
w-011-14 w-037-14	0.97616	0.99991	w-023-1 w-041-1	0.95414	0.98318
w-014-14 w-036-14	0.76624	0.99999	w-034-14 w-038-14	0.99998	0.99922
w-015-1 w-048-1	0.99999	0.98557	w-034-14 w-052-14	0.99999	0.98672

Figura 4.32: Risultati dei matching tra i morphing dei software Morphthng (a) e Sqrlz Morph (b) con Luxand Face Mather

Anche in questo test la percentuale di successo dei software rimane sempre all'interno dello stesso range; per Morhthing gli esiti positivi sono stati del 54%, mentre per Sqirlz Morph sono del 62%.

Questo significa che per il 55% dei casi gli algoritmi allo stato dell'arte dei sistemi di riconoscimento non rilevano le alterazioni digitali delle immagini, riconoscendo nell'immagine alterata il soggetto raffigurato.

Conclusioni

Gli esiti dei test svolti per questa tesi mostrano l'esistenza di una discreta possibilità di successo di ingannare un sistema di controllo basato sul riconoscimento del volto, come i sistemi ABC che si stanno diffondendo nelle frontiere aeroportuali.

Anche se le percentuali di successo si aggirano attorno al 55%, nel caso di un sistema di sicurezza questo tipo di risultati comporta un notevole rischio.

Il problema del possibile utilizzo da parte di due persone dello stesso passaporto, nasce quando una delle due persone è un ricercato.

Non potendo oltrepassare un sistema di identificazione umano o in questo caso elettronico, ha la possibilità di far richiedere ad un altro soggetto, non ricercato, il passaporto elettronico utilizzando come fototessera un'immagine *morphed* dei due soggetti.

In questo modo l'ente rilasciante il passaporto può accettare la fototessera come caratteristica biometrica e rilasciare il passaporto, avendo identificato nell'immagine il soggetto che ha richiesto il documento.

Rilasciando il passaporto l'ente ha dichiarato valido il documento che sarà in grado di superare un controllo elettronico.

Il criminale a questo punto essendo identificato in un passaporto valido ha la capacità di oltrepassare un sistema di controllo, con una probabilità che secondo questi test si aggira attorno al 55% se si tratta di un sistema di riconoscimento elettronico.

Significa che un criminale su due ha la possibilità di oltrepassare un sistema di controllo ABC.

Nel caso di un controllo d'identità svolto da un agente di frontiera la circostanza rimane invariante; le immagini prodotte dal *morphing* mostrano un volto realistico in cui difficilmente non si identificano i soggetti che hanno preso parte al *morphing*, quindi anche un agente esperto addetto al riconoscimento di identità può essere facilmente ingannato.

Il metodo più efficace per risolvere questo problema è evitare di dare al cittadino la possibilità di fornire la fototessera per il documento di passaporto elettronico, e

effettuare l'acquisizione dell'immagine live, al momento del rilascio del documento, usando una fotocamera ad alta qualità in un ambiente controllato.

Un altro caso possibile sarebbe quello di fornire alle autorità di emissione dei documenti, qualora la fototessera dovesse essere fornita dal cittadino, un software in grado di riconoscere la presenza di alterazioni in un'immagine.

Un sistema di riconoscimento del volto potrebbe essere utilizzato per l'avvertimento in caso di bassi punteggi di corrispondenza tra la fototessera fornita dal cittadino e un'altra immagine acquisita.

Bibliografia e Sitografia

- [1] *International Civil Aviation Organization (ICAO)*, <http://www.icao.int/Security/mrtd/Pages/24-NOV-2015.aspx>
- [2] *Associazione Nazionale Fotografi Professionisti*, <http://www.fotografi.org/fototessera/>
- [3] *Wikipedia*, <http://it.wikipedia.org/wiki/Morphing>
- [4] Beier T., Costa B., Darsa L., Velho L. (1997), *Warping and Morphing of Graphical Objects*, Siggraph
- [5] Li S. Z., Jain A.K. (2004), *Handbook of Face Recognition*, Springer
- [6] Brand J., Mason J.S. (2000) *A Comparative Assessment of Three Approaches to Pixel-level Human Skin-Detection*
- [7] *Wikipedia*, http://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
- [8] *Wikipedia*, http://en.wikipedia.org/wiki/Haar-like_features
- [9] *Wikipedia*, <http://en.wikipedia.org/wiki/AdaBoost>
- [10] *Università degli Studi di Bari Aldo Moro*, <http://www.di.uniba.it/~ig/GaborFilter/html/filtro.html>
- [11] Pentland A., Turk M. (1991), *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience
- [12] *Wikipedia*, http://it.wikipedia.org/wiki/Vladimir_Vapnik
- [13] Heisele B., Ho P., Poggio T., *Face Recognition with Support Vector Machines: Global versus Component-based Approach*, Massachusetts Institute of Technology
- [14] Lazzerini B., *Introduzione alle Reti Neurali*, Dipartimento di Ingegneria della Informazione
- [15] Nefian A.V., Hayes M.H. (1998), *A Hidden Markov Model for face recognition*
- [16] Nefian A.V., Hayes M.H. (1999), *An Embedded HMM – Based Approach For Face Detection and Recognition*, Georgia Institute of Technology

- [17] *Gimp – The GNU Manipulation Program*, <http://www.gimp.org/>
- [18] *Morphthing.com*, <http://www.morphthing.com/>
- [19] *Sqirlz Morph*, <http://www.xiberpix.net/SqirlzMorph.html>
- [20] *FaceMorpher*, <http://www.facemorpher.com/>
- [21] *Face On Face*, <http://faceonface.net/>
- [22] *Neurotechnology*, <http://www.neurotechnology.com/>
- [23] *Luxand*, <https://www.luxand.com/>
- [24] Wiskott L., Von Der Malsburg C. (1995), *Recognizing Faces by Dynamic Link Matching*
- [25] Martinez A. M., Benavente R. (1998), *The AR Face Database*, Computer Vision Center, CVC Technical Report
- [26] Ferrara M., Franco A., Maltoni D., Sun Y. (2011), *On the impact of Alterations on Face Photo Recognition Accuracy*, Department of Computer Science and Engineering, University of Bologna, Cesena
- [27] *Lift Magic*, <http://www4.liftmagic.com/liftmagic/liftupload.php>
- [28] Ogihara A., Shintani A., Takamatsu S., Igawa S. (1996), *Speech Recognition based on the Fusion of Visual and Auditory Information using Full-Frame Colour image*, IEICE Trans. Fundamentals
- [29] Karande K.J., Talbar S.N. (2008), *Face Recognition under Variation of Pose and Illumination using Independent Component Analysis*

Elenco delle figure

1.1: Esempio di e-Passport Gates	2
2.1: Dissolvenza incrociata fra due immagini	6
2.2: Domain transformation di un'immagine	7
2.3: Range transformation di un'immagine.....	8
2.4: Esempi di trasformazione affine: traslazione, rotazione, riscalatura, affine.	9
2.5: Esempi di proiezione prospettica a sinistra e proiezione parallela a destra	10
2.6: Trasformazione con interpolazione affine.....	11
2.7: Trasformazione con interpolazione bilineare	12
2.8: Immagine di una scacchiera.....	13
2.9: Confronto fra le tre tecniche, a partire dall'alto: affine, bilineare e proiettiva.....	14
2.10: Esempio di <i>mesh warping</i>	15
2.11: Fasi dell'algoritmo di <i>Mesh Warping</i>	16
2.12: <i>Feature-Based Warping</i> con una linea	17
2.13: <i>Feature-Based Warping</i> con più segmenti.....	19
2.14: Esempio di <i>Feature-Based Warping</i> con due linee	20
2.15: <i>Thin Plate Spline</i>	21
2.16: Esempio di posizionamento dei <i>landmarks</i>	23
2.17: <i>Warping</i> che deforma il viso di Bush per assomigliare a quello di Schwarzenegger.....	23
2.18: <i>Warping</i> inverso della foto di Schwarzenegger per assomigliare a quella di Bush	24
2.19: <i>Face Morphing</i> tra la foto di Bush e quella di Schwarzenegger	24
3.1: Esempio di sistema di localizzazione di volti	27
3.2: Grafici del livello di rosso vs. il livello di verde per: a) immagini di pelle b) immagini di non-pelle	29
3.3: <i>Feature Haar-like</i>	29

3.4: Pixel presi in considerazione per il calcolo dei nuovi pixel dell' <i>immagine integrale</i>	30
3.5: Cascata di classificatori.....	32
3.6: (a) funzione sinusoidale, (b) kernel Gaussiano, (c) filtro di Gabor	35
3.7: Filtro di Gabor, a sinistra la parte reale e a destra la parte immaginaria del filtro	36
3.8: Esempio di facce utilizzate in un training set (a) e le relative <i>eigenfaces</i> (b)	37
3.9: Esempio di rete di Hopfield.....	42
3.10: Esempio di rete feed-forward	42
3.11: Esempio di rete ricorrente.....	43
3.12: Suddivisione di un viso in stati	47
4.1: Esempi di alterazioni di immagini. (a) Immagine Originale. (b) <i>Barrel Distorsion</i> . (c) <i>Vertical Contraction</i> . (d) <i>Vertical Extension</i> . (e) <i>Digital Beautification</i> . (f) Differenza dei pixel tra l'immagine originale (a) e alterata tramite <i>digital beautification</i> (e)	51
4.2: Home Page di Morphthing	53
4.3: Esempio landmarks di MorphThing.....	54
4.4: Finestra Images di Fotomorph	56
4.5: Finestra Control di Fotomorph	56
4.6: Finestra di Sqirlz Morph.....	57
4.7: Ritocco manuale in FaceMorpherLite	58
4.8: Guida di FaceMorpherLite	59
4.9: Finestra del posizionamento dei landmarks in Faceonface	60
4.10: Tabella di confronto tra i cinque software di morphing.....	61
4.11: immagini di partenza per la realizzazione del morphing.....	61
4.12: Risultati di morphing da in alto a sinistra a destra di: Morphthing, FotoMorph, FaceonFace, Sqirlz Morph, Face Morpher Lite.....	62
4.13: Immagini prelevate dal database AR. A sinistra la posa 1 e a destra la posa 14	64
4.14: Esempio di immagine morphed non ritoccata a sinistra, e ritoccata a destra.....	65

4.15: Face detection di Verilook 5.3/MegaMatcher.....	66
4.16: Matching con Luxand SDK Face Matcher	67
4.17: Risultati dei matching tra i morphing del software Morphthing	69
4.18: Risultati dei matching tra i morphing del software Fotomorph	69
4.19: Risultati dei matching tra i morphing del software Sqirl Morph.....	70
4.20: Risultati dei matching tra i morphing del software Face Morpher Lite	70
4.21: Risultati dei matching tra i morphing del software Face on Face.....	71
4.22: Morphing N°2 Morphthing.....	72
4.23: Morphing N°10 Morphthing	72
4.24: Morphing N°6 Fotomorph	73
4.25: Morphing N°9 Fotomorph	73
4.26: Morphing N°7 Sqirlz Morph.....	73
4.27: Morphing N°15 Sqirlz Morph	73
4.28: Morphing N°4 Face Morpher Lite	74
4.29: Morphing N°18 Face Morpher Lite.....	74
4.30: Morphing N°1 Face on Face.....	74
4.31: Morphing N°8 Face on face	74
4.32: Risultati dei matching tra i morphing dei software Morphthing (a) e Sqirlz Morph (b) con Luxand Face Matcher.....	76