Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

AUTOVETTORI E RICONOSCIMENTO FACCIALE

Tesi di Laurea in Geometria

Relatore:
Chiar.mo Prof.
MASSIMO FERRI

Presentata da:
MARIA CHIARA BAGLI

II Sessione
2015/2016

...a mia nipote Giada

Indice

Elenco delle figure

Introduzione

Le tecniche di automatizzazione del riconoscimento facciale sono finalizzate all'identificazione di soggetti presenti in foto, filmati o supporti digitali dato un database iniziale di volti in condizioni variabili di luce, espressione o rotazione della testa. Questo problema interessa svariati ambiti tra cui quello della sicurezza e quello della manipolazione di immagini e filmati digitali.

La tesi tratta della tecnica per il riconoscimento facciale delle autofacce, seguendo come traccia l'articolo "Eigenface for Recognition" di Turk e Pentland, pubblicato nel 1991. In particolare ho verificato la semplicità dell'algoritmo che caratterizza questa tecnica testandola sulle immagini di alcuni soggetti dell'MR2 Face Database grazie al codice che ho implementato a partire da quello di Michael Scheinfeild.

Il primo capitolo tratta brevemente della storia delle tecniche di riconoscimento facciale studiate fino agli anni '90.

Nel secondo capitolo vengono riportati alcuni richiami di autovalori, autovettori, varianza e covarianza.

Nel terzo capitolo viene trattata la tecnica autofacce passaggio per passaggio, facendo riferimento alla verifica che ho fatto in laboratorio. In particolare si trattano la generazione dello spazio delle autofacce e quella della procedura di riconoscimento facciale, seguite da un breve accenno al tipo di problema a cui si approccia questa tecnica.

Capitolo 1

Introduzione al riconoscimento facciale

1.1 Tentativi di soluzione fino agli anni '90

A partire dalla seconda metà del ventesimo secolo si iniziò a cercare un modo di automatizzare il riconoscimento facciale. Tra alcuni dei tentativi più importanti vengono riportati i seguenti:

- **1964-1965** Negli anni sessanta Woody Bledsoe, Helen Chan e a Charles Bisson svilupparono il primo sistema semi-automatico di riconoscimento facciale. Questo richiedeva un soggetto che localizzasse tramite una tavola grafica, detta GRAFACON o RANDTABLET, delle caratteristiche predefinite (come centro delle pupille, orecchie, punto più esterno e più interno dell'occhio) sulle fotografie. Fatto ciò venivano calcolate dal sistema 20 distanze tra le varie caratteristiche (distanza tra pupilla e pupilla etc). Ogni soggetto nel database inizale veniva così associato alla lista di queste distanze. Durante la fase di riconoscimento veniva fatto lo stesso e il nuovo insieme di distanze veniva comparato con gli insiemi di distanze nel database iniziale e veniva selezionato il soggetto appartenente a questo le cui distanze associate erano più simili. Woody Bledsoe disse a proposito delle difficoltà di questo sistema.
  "This recognition problem is made difficult by the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. Some other attempts at facial recognition by machine have allowed for little or no variability in these quantities. Yet the method of correlation (or pattern matching) of unprocessed optical data, which is often used by some researchers, is certain to fail in cases where the

variability is great. In particular, the correlation is very low between two pictures of the same person with two different head rotations."[11]

- **1971** Nel 1971 Goldstein, Harmon e Lesk svilupparono un sistema che automatizzava il riconoscimento facciale grazie a 21 segni che corrispondevano a delle caratteristiche del volto. Anche questi segni venivano inseriti manualmente. Gli autori scrissero in fondo al loro articolo in cui veniva presentato questa innovativa tecnica:
  "Work in progress uses human-computer dialog for face recognition and file retrieval. In it we replace the techniques reported here (feature-extremeness ranking and binary sorting) by computations of distances between a target description and the feature vectors of the population. We shall use this to refine our techniques for effective human-face recognition and to develop adaptive real-time man-machine interrogation."[5]

- **1986** Kirby e Sirovich applicarono l'analisi in componenti principali (PCA), conosciuta anche come trasformata di Karhunen-Loève, al problema del riconoscimento facciale. Fu considerata una pietra miliare in questo campo. Presentarono così il loro articolo:
  "In brief, we demonstrate that any particular face can be economically represented in terms of a best coordinate system that we term eigenpicture[...]Thus, in principle, any collection of faces could be classified by storing a small collection of numbers for each face and a small set of standard pictures known as eigenpictures" [9]

- **1991** Partendo dallo studio di Kirby e Sirovich, Turk e Pentland scoprirono che, usando la tecnica delle eigenpictures, si potevano rilevare le facce nelle immagini, in particolare quindi si poteva ricondurre il soggetto di una fotografia ad uno eventualmente presente in un database iniziale. Con quelle che loro chiamarono *eigenfaces* e che potremmo definire autofacce, si ha l'inizio di una tecnica usata ancora oggi per il riconoscimento facciale.
  "Our goal, which we believe we have reached, was to develop a computational model of face recognition that is fast, reasonably simple, and accurate in constrained environments such as an office or a household. In addition the approach is biologically implementable and is in concert with preliminary findings in the physiology and psychology of face recognition"[12]

Capitolo 2

Richiami preliminari

2.1 Autovettori e autovalori

Definizione 2.1 (Matrice)**.** Sia X un insieme e sia (m, n) una coppia di interi positivi. Si dice *matrice di tipo* $m \times n$ *a coefficienti in* X un'applicazione $A : \mathbb{N}_m \times \mathbb{N}_n \to X$. Se $m = n$ allora A è detta *matrice quadrata di ordine* n. Una generica matrice A di tipo $m \times n$ a coefficienti in X è dunque completamente individuata dagli $m \cdot n$ elementi $A(i, j) \in X$,con $i \in \mathbb{N}_m$ e $j \in \mathbb{N}_n$.
Per ogni $(i, j) \in \mathbb{N}_m \times \mathbb{N}_n$, porremo $A(i, j) = a_{i,j}$.

$$\boldsymbol{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \dots & a_{1,n} \\ a_{2,1} & a_{2,2} \dots & a_{2,\mathrm{n}} \\ & \vdots & \\ a_{\mathrm{m},1} & a_{\mathrm{m},2} \dots & a_{\mathrm{m,n}} \end{pmatrix}$$

Scriveremo anche $\boldsymbol{A} = (a_{\mathrm{i,j}})$. Siano v_i matrici di tipo $m \times 1$ $\forall 1 \leq i \leq n$ che chiameremo vettori colonna. Scriveremo

$$\boldsymbol{B} = [v_1, v_2, \cdots, v_n]$$

per indicare la matrice di tipo $m \times n$ che ha come colonne i vettori v_i.

Definizione 2.2 (Trasposta)**.** La trasposta della matrice $A = (a_{i,j})$ di tipo $m \times n$ è la matrice di tipo $n \times m$

$$\boldsymbol{A^T} = \begin{pmatrix} a_{1,1} & a_{2,1} \dots & a_{\mathrm{m},1} \\ a_{1,2} & a_{2,2} \dots & a_{\mathrm{m},2} \\ & \vdots & \\ a_{1,\mathrm{n}} & a_{2,\mathrm{n}} \dots & a_{\mathrm{m,n}} \end{pmatrix}$$

ottenuta scambiando tra loro le righe e le colonne di A.

Definizione 2.3 (Matrice simmetrica)**.** Una matrice quadrata A di ordine n è detta *simmetrica* se $A^T = A$

Definizione 2.4 (Autovalori e autovettori)**.** Sia $\mathbb{K}$ campo. Uno scalare $\lambda \in \mathbb{K}$ è detto *autovalore* dell'operatore lineare T su V, dove V è uno spazio vettoriale su $\mathbb{K}$, se il sottospazio $U_\lambda(T)$ di V definito come:

$$U_\lambda = U_\lambda(T) = \{v \in V \mid T(v) = \lambda \cdot v\}$$

non si riduce al solo vettore nullo 0 di V. In tal caso, $U_\lambda(T)$ è detto *autospazio* di T, relativo all'autovalore λ; gli elementi di $U_\lambda(T)$ sono detti *autovettori* di T relativi all'autovalore λ.

Proposizione 2.1.1. *Sia X un insieme e sia (m, n) una coppia di interi positivi. Sia A una matrice di tipo $n \times m$ a coefficienti in X e sia A^T la sua trasposta. Il prodotto AA^T è una matrice quadrata di ordine m simmetrica.*

Proposizione 2.1.2. *Siano A, B due matrici quadrate simmetriche di ordine n tali che $\exists \mu \in \mathbb{R}$ tale che $A = \mu B$. Allora u è un autovettore di $A \iff$ lo è di B.*

Corollario 2.1.3. *Sia A una matrice di tipo $m \times n$, B matrice quadrata simmetrica di ordine m e $\mu \in \mathbb{R}$ tale che $B = \frac{1}{\mu} AA^T$. Allora u è autovettore di $B \iff$ lo è di AA^T.*

Proposizione 2.1.4. *Sia A una matrice di tipo $m \times n$. Se u è un autovettore di $A^T A$ allora Au è un autovettore di AA^T.*

Dimostrazione. Sia u un arbitrario autovettore $m \times 1$ di $A^T A$.

$$\exists \lambda \in \mathbb{R} \quad t.c. \quad A^T A u = \lambda u$$

$$\implies AA^T Au = A\lambda u$$

$$\implies AA^T Au = \lambda Au$$

$$\implies (AA^T)Au = \lambda Au$$

Poichè è mostrato per un arbitrario autovettore u di $A^T A$ che Au di tipo $n \times 1$ è un autovettore di AA^T, allora lo abbiamo mostrato per ogni autovettore di $A^T A$. □

2.2 Varianza e covarianza

Definizione 2.5 (Variabile aleatoria discreta)**.** Una variabile aleatoria discreta X è una funzione

$$X : \Omega \longrightarrow \mathbb{R}^d$$

con $d \in \mathbb{N}$, Ω insieme finito e P misura di probabilità.

Definizione 2.6 (Valore atteso)**.** Sia $X : \Omega \longrightarrow \mathbb{R}^d$ variabile aleatoria discreta. Il valore atteso di X è

$$E[X] = \sum_{\omega \in \Omega} X(\omega) P(\{\omega\})$$

Definizione 2.7 (Varianza)**.** Sia X una variabile aleatoria a valori in $\mathbb{R}$. La varianza di X é

$$var(X) = E[(X - E[X])^2]$$

Definizione 2.8 (Matrice di covarianza)**.** Sia $X = (X_1, \ldots, X_d)$ una variabile aleatoria a valori in $\mathbb{R}^d$. La matrice di convarianza di X è la matrice $d \times d$ simmetrica

$$Cov(X) = E[(X - E[X])(X - E[X])^T]$$

dove $(X - E[X])$ è una matrice $d \times 1$.
Si può anche scrivere $Cov(X) = (c_{i,j})$ con $c_{i,j} = E[(X_i - E[X_i])(X_j - E[X_j])]$.

$$Cov(X) = \begin{pmatrix} c_{1,1} & c_{2,1} \ldots & c_{d,1} \\ c_{2,1} & c_{2,2} \ldots & c_{d,2} \\ & \vdots & \\ c_{\mathrm{d},1} & c_{\mathrm{d},2} \ldots & c_{d,d} \end{pmatrix}$$

2.3 PCA

Per studiare lo spazio delle eigenfaces è necessario l'uso dell'analisi in componenti principali (PCA). Considerate delle variabili e i valori da esse assunti, lo scopo del PCA è quello di ridurre il numero di queste variabili,"eliminando" quelle a cui è associata una minor varianza e "conservando" così le variabili che meglio descrivono la dispersione tra i vari dati.
Consideriamo un sistema di coordinate iniziale in cui si trovano dei dati. Quello che vogliamo è costruire un nuovo sistema di coordinate ruotando quello precedente. La variabile con maggior varianza viene proiettata sul primo asse del nuovo sistema mentre le altre variabili, in ordine decrescente di varianza, ruotano di conseguenza per rimanere ortogonali agli assi trovati in precedenza. Questi nuovi assi sono chiamati **componenti principali**. Procedendo in questo modo, le ultime componenti principali trovate descriveranno solo una minima percentuale di dispersione dei dati. Sono queste che vengono eliminate, riducendo così la dimensione del sistema. Viene riportato nella figura 2.1 un esempio di PCA in due dimensioni che rende evidente questo processo. Si può notare che il nuovo asse PCA1 è quello parallelo alla variabile che massimizza la varianza. [3]
Per calcolare le componenti principali è possibile anche utilizzare la matrice

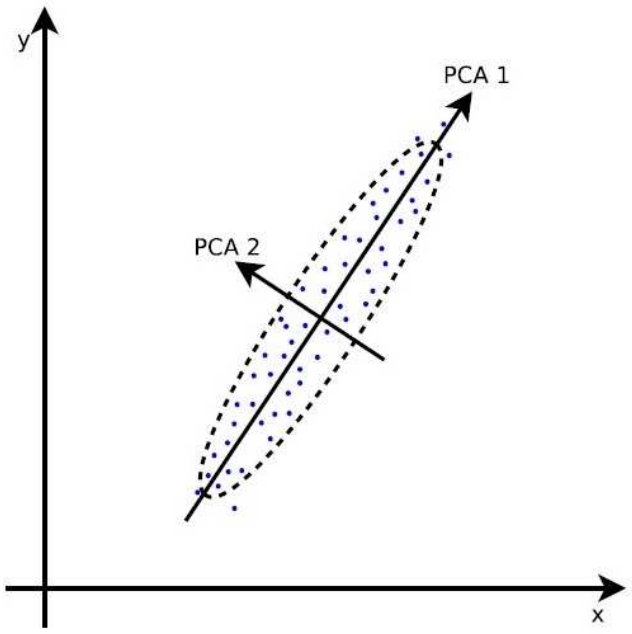

Figura 2.1: PCA in due dimensioni

di covarianza della variabile ed è questo il metodo sfruttato nella costruzione delle autofacce.

Capitolo 3

Lo spazio delle autofacce

In questo capitolo riporto i passaggi della costruzione dello spazio delle facce e dell'utilizzo di esso per il riconoscimento facciale. Il codice che ho implementato seguendo la traccia di quello di Michael Scheinfeild e che ho usato per la procedura è presente nell'appendice A. Il database che ho considerato è parte del MR2 FACE DATABASE[10]. Il capitolo segue la traccia dell'articolo di Mark Turk e Alex Pentland, "Eigenfaces for recognition" pubblicato nel 1991 sul *Journal of Cognitive Neuroscience.*[12].

3.1 Generazione dello spazio

Consideriamo un database di immagini ognuna raffigurante il volto di un soggetto. *In generale nei database ci possono essere più facce di uno stesso soggetto. In questo caso le facce di una stessa persona danno vita ad una classe. Il procedimento che andremo a vedere non cambia in quanto nel caso delle classi si considera la faccia media (vedi 3.3) di una classe.* Le immagini dovrebbero avere le stesse condizioni di luce ed i volti in esse raffigurati dovrebbero essere allineati.
Nel nostro caso consideriamo un database di 70 soggetti, ognuno presente una sola volta nelle immagini. Le immagini hanno tutte sfondo bianco e gli occhi e la bocca allineati. In origine le foto hanno dimensione 3120×3120 pixels e sono a colori. Le ho ridotte a 256×256 pixels e trasformate in scala di grigi per ridurre il costo computazionale dell'implementazione. In figura 3.1 viene riportata la foto di un soggetto che è stata riscalata e resa in bianco e nero. Si tenga conto che le due immagini non mantengono le proporzioni iniziali in quanto la più grande non potrebbe essere contenuta nel foglio oppure la seconda sarebbe troppo piccola. In figura 3.2 viene riportata parte del database così costruito. Ogni immagine è rappresentata da una matrice

Figura 3.1: Immagine riscalata e trasformata in scala di grigi

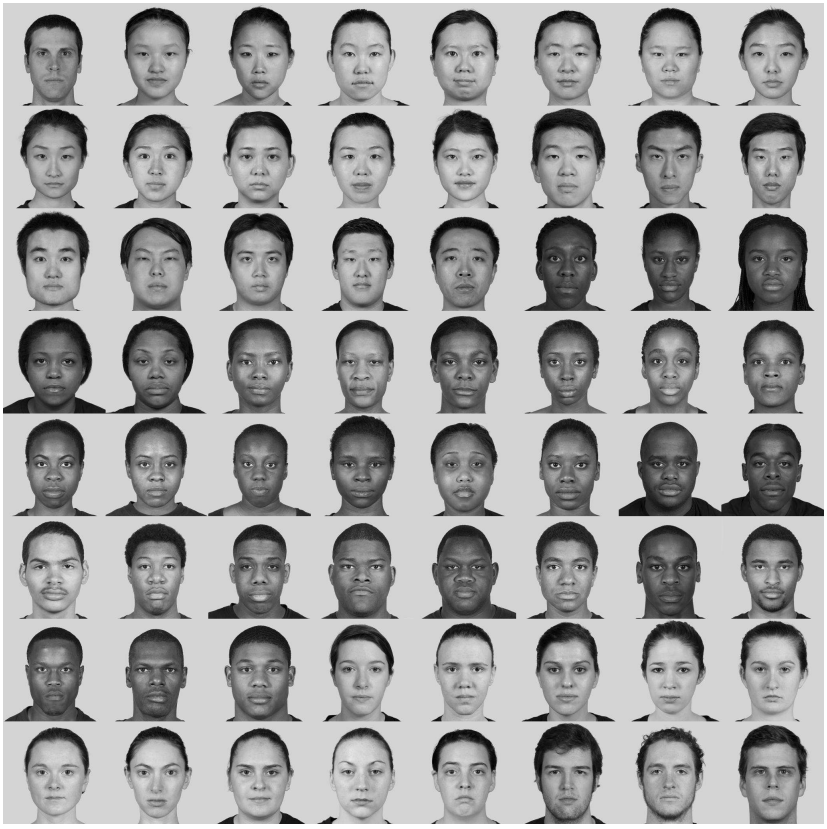

Figura 3.2: Database iniziale

di tipo $n \times m$ dei valori dei pixels. Nel nostro caso abbiamo una matrice quadrata di ordine 256 ed ogni elemento riporta un valore da 0 a 255 in quanto tutte le immagini sono stase rese in bianco e nero.

Sia I_1 l'immagine del soggetto mostrata precedentemente. Sia $n = 256$, $M = 70$ dove M è il numero delle immagini nel nostro database.[6] Avremo allora:

$$\boldsymbol{I_1} = \begin{pmatrix} p_{1,1}\ p_{2,1} \cdots p_{\mathrm{n},1} \\ p_{1,2}\ p_{2,2} \cdots p_{\mathrm{n},2} \\ \vdots \\ p_{1,\mathrm{n}}\ p_{2,\mathrm{n}} \cdots p_{\mathrm{n},\mathrm{n}} \end{pmatrix}$$

Ora vogliamo rappresentare I_1 come un vettore che chiamiamo Γ_1. Costruiamo Γ_1 concatenando le colonne di I_1. Abbiamo quindi:

$$\boldsymbol{\Gamma_1} = \begin{pmatrix} p_{1,1} \\ p_{2,1} \\ \cdot \\ \cdot \\ \cdot \\ p_{\mathrm{i},1} \\ p_{n,1} \\ p_{1,2} \\ p_{2,2} \\ \cdot \\ \cdot \\ \cdot \\ p_{n,2} \\ p_{1,3} \\ \cdot \\ \cdot \\ \cdot \\ p_{n,n} \end{pmatrix}$$

Applicando questa trasformazione a tutte le M immagini del nostro database, otteniamo un nuovo insieme S:

$$S = \{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}$$

D'ora in poi ci riferiremo alle immagini considerando i vettori colonna ad esse associati. Quello che ci interessa è considerare le caratteristiche che differenziano un soggetto dai restanti del database. Calcoliamo quindi la *faccia media* che identifichiamo con Ψ:

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i \tag{3.1}$$

Figura 3.3: Faccia media

Come possiamo notare in figura 3.3 ciò che abbiamo chiamato faccia media, rappresentata dal vettore Ψ sembra effettivamente un nuovo volto. La parte più definita dell'immagine è quella rappresentante gli occhi. Questo accade perchè i soggetti, pur essendo notevolmente diversi tra loro, hanno una certa somiglianza in questa parte.
Riconsideriamo ora Γ_1 e sottraiamo ad essa la faccia media. In figura 3.4 è riportato il risultato ottenuto.

$$\Phi_1 = \Gamma_1 - \Psi$$

Ripetendo il procedimento per ogni faccia, al fine di concentrarci sulle ca-

Figura 3.4: Faccia senza media

ratteristiche che rendono particolari i vari soggetti (3.2), otteniamo un nuovo database che possiamo vedere in parte in figura 3.5.

$$\Phi_i = \Gamma_i - \Psi \quad \forall 1 \leq i \leq M \tag{3.2}$$

Figura 3.5: Database delle facce senza media

Consideriamo il nuovo insieme di vettori $\Psi_i \ \forall 1 \leq i \leq M$ che possiamo rappresentare come la matrice A di tipo $n^2 \times M$:

$$A = [\Phi_1, \Phi_2, \cdots, \Phi_M]$$

Abbiamo ora una collezione di dati che costituirà il nostro training set e che definisce lo *spazio delle facce*. Questo spazio ha per ora dimensione n^2. Il nostro scopo è ridurla mantenendo intatte le informazioni di dispersione dei dati. Questo insieme di vettori sarà soggetto all'analisi in componenti principali.
Costruiamo la matrice quadrata di covarianza, di ordine n^2.

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T$$

Si può dimostrare che $C = \frac{1}{M} AA^T$. [6]
Come abbiamo visto dobbiamo cercare le componenti principali dello spazio. Per far questo possiamo cercare gli autovettori di C. Essendo questa una matrice quadrata di ordine n^2 per farlo è necessario un costo computazionale elevato. Ma C è matrice delle covarianza quindi simmetrica e, grazie al corollario 2.1.3, possiamo cercare i suoi autovettori cercando quelli di AA^T. Per la proposizione 2.1.4 se u è un autovettore di $A^T A$ allora Au è un autovettore di AA^T. Calcoliamo quindi gli autovettori di $A^T A$ con un minor costo computazionale in quanto è una matrice di tipo $M \times M$ e successivamente calcoliamo gli autovettori di AA^T. Dopo aver calcolato questi autovalori di AA^T, quindi di C, li ordiniamo in ordine decrescente a seconda degli autovettori ad essi associati. Gli autovettori (che da ora chiameremo anche *autofacce*) associati agli autovalori più grandi, saranno quelli che caratterizzano in maggior parte la varianza del training set. In figura 3.6 è mostrata la maggior parte delle autofacce così calcolata. Come si può notare le ultime autofacce rappresentate sono quasi tutte nere. Questo perchè corrispondono agli autovalori più piccoli quindi non apportano un grande contributo alla caratterizzazione dei soggetti.

3.1.1 Riduzione della dimensione

Ora passiamo all'effettiva riduzione della dimensione dello spazio delle facce. Per prima cosa si sceglie un valore di soglia sotto al quale verranno eliminate le autofacce; La soglia si può scegliere in base alla percentuale di varianza totale che si vuol rappresentare.
Supponiamo ad esempio di volerne rappresentare il 95%.
Sia η la somma di tutti gli autovalori. Consideriamo l'autovalore più grande di C, λ_1. Nel caso $\frac{\lambda_1}{\eta} < 0.95$ sommiamo a λ_1 il secondo autovalore più grande, λ_2. Nel caso $\frac{\lambda_1+\lambda_2}{\eta} < 0.95$ sommiamo il terzo autovalore e così via. Non appena per un autovalore λ_i si ha che

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_i}{\eta} \geq 0.95$$

ci fermiamo. i è il numero di autofacce necessarie a descrivere almeno il 95% della varianza totale. Il nuovo spazio così costruito è di dimensione ridotta pur mantenendo le informazioni più importanti riguardo la varianza. Nella figura 3.7 viene mostrata parte del nuovo spazio. Nel caso del nostro database il numero di autofacce (che ricordiamo essere rappresentate dagli autovettori), calcolato grazie al metodo visto in precedenza, è 46. Questo valore è molto minore rispetto a $n^2 = 65536$ ed è dell'ordine del numero di

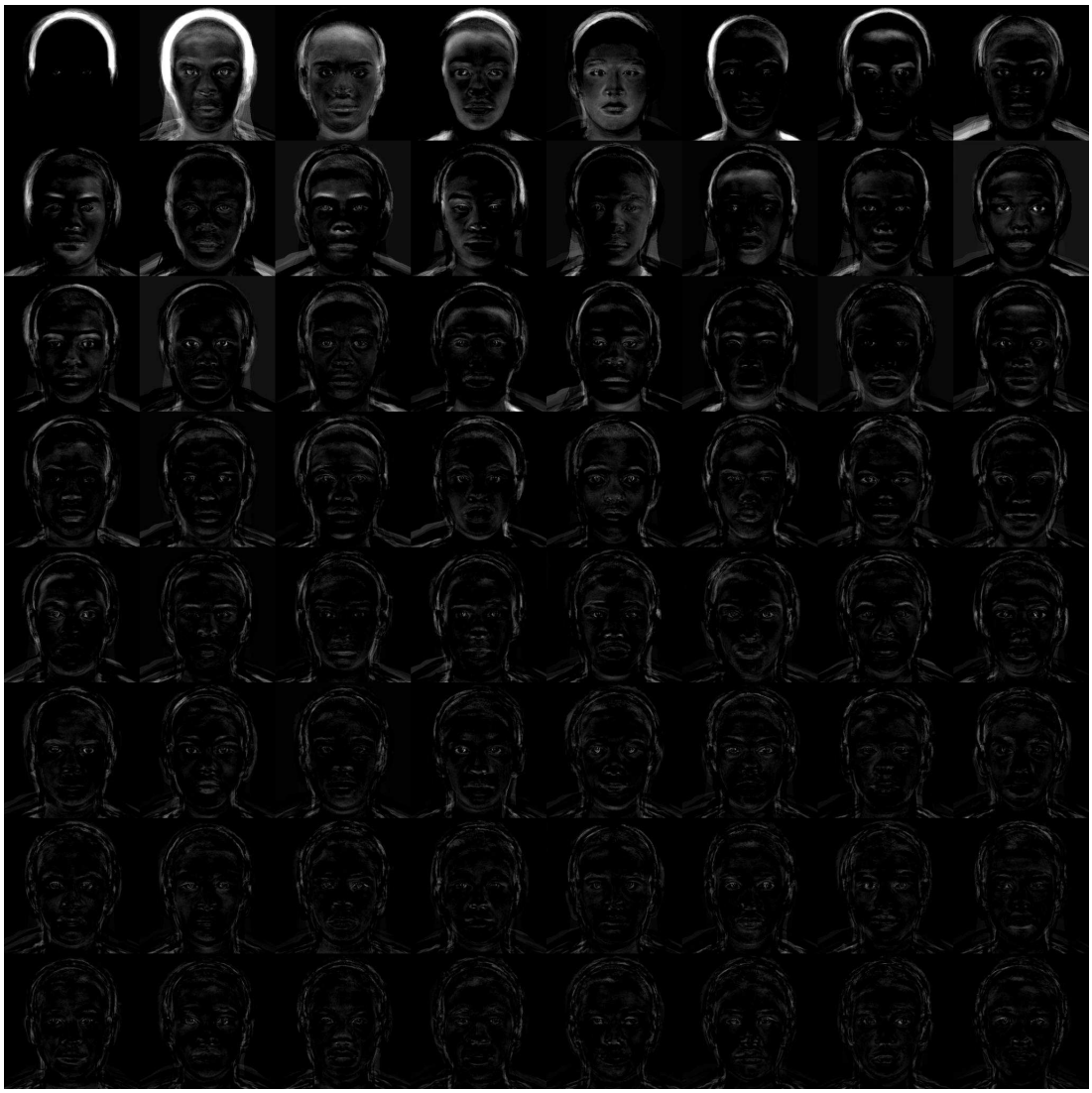

Figura 3.6: Autofacce senza riduzione

soggetti nel database iniziale.
È possibile notare che tutte le facce autofacce presenti nella figura 3.7 sono ben definite e che quelle che in figura 3.6 erano quasi o completamente nere sono state eliminate.
Nell'appendice B viene riportata la tabella degli autovalori dove i primi 46 sono visibili in grassetto.

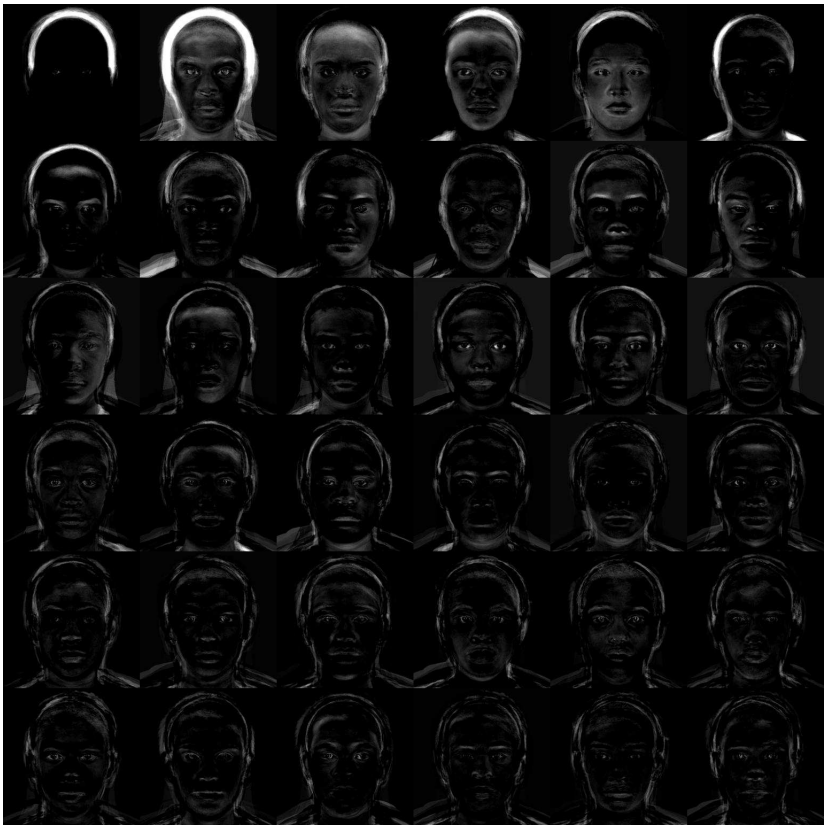

Figura 3.7: Autofacce

3.1.2 Proiezione delle facce

A questo punto possiamo prendere le facce del database iniziale e proiettarle nel nuovo spazio in modo che ognuna di esse sia combinazione lineare delle 46 autofacce. Consideriamo ad esempio Φ_1 che corrisponde al primo soggetto, privato della faccia media e presente nel database iniziale, che vogliamo proiettare nel nuovo spazio. Siano $u_k \quad \forall 1 \leq k \leq 46$ gli autovettori corrispondenti alle autofacce. Avremo $a_{k,1} = u_k^T \Phi_1$, cioè:

$$a_{k,1} = \begin{pmatrix} u_{k_1} & u_{k_2} & \cdots & u_{k_n^2} \end{pmatrix} \begin{pmatrix} \Phi_{1_1} \\ \Phi_{1_2} \\ \cdot \\ \cdot \\ \Phi_{1_{n^2}} \end{pmatrix}$$

Dove $a_{k,1}$ è il k-esimo coefficiente di proiezione del primo soggetto, $n = 256$ e u_{k_i} è l'i-esima componente di $u_k \quad \forall 1 \leq i \leq 65536$. A questo punto la pro-

iezione di Φ_1 potrà essere scritta come combinazione lineare delle autofacce, ovvero come:

$$a_{1,1}u_1 + a_{2,1}u_2 + \cdots + a_{46,1}u_{46}$$

In figura 3.8 é possibile vedere la proiezione nello spazio delle autofacce di Φ_1.

Figura 3.8: Proiezione di Φ_1

Ripetendo questo procedimento per ogni faccia del nostro database iniziale otterremo la loro posizione all'interno dello spazio delle facce e potremo così iniziare il riconoscimento facciale. La matrice dei coefficienti di proiezione di tutte le facce sarà quindi

$$\begin{pmatrix} a_{1,1} & a_{1,2} \dots & a_{1,70} \\ a_{2,1} & a_{1,3} \dots & a_{2,70} \\ & \vdots & \\ a_{46,1} & a_{1,4} \dots & a_{46,70} \end{pmatrix}$$

Perchè 46 è il numero di autofacce e 70 è il numero di facce presenti nel database. É possibile vedere la matrice dei coefficienti di proiezione in appendice C. Le autofacce presenti nello spazio delle facce assomigliano a dei fantasmi e per questo motivo vengono anche chiamate *ghostfaces*.

3.2 Riconoscimento facciale

In questa sezione viene mostrato come si possono usare le autofacce per il problema del riconoscimento facciale.
Abbiamo appena visto che, grazie alla proiezione delle immagini del database iniziale nello spazio delle facce, studiamo di fatto la loro posizione nel nuovo

sistema di coordinate. In sostanza, per capire se un nuovo soggetto è presente nel database iniziale consideriamo la sua proiezione nel nuovo spazio e calcoliamo la distanza rispetto alle altre facce. Nel caso sia abbastanza vicino ad una di esse avviene il riconoscimento. Mostriamo il procedimento. Per prima cosa scegliamo arbitrariamente una soglia ovvero un valore sotto al quale la nuova immagine sarà considerata "abbastanza" vicina al nuovo spazio. La scelta della soglia dipende dall'obiettivo che si vuol raggiungere. I due obiettivi più comuni sono:

- Riconoscere una faccia in un database iniziale
- Riconoscere se un'immagine rappresenta una faccia

Il nostro scopo è quello di capire se una nuova faccia è presente nel database iniziale quindi consideriamo il primo caso. Scegliamo arbitrariamente un valore di soglia, ad esempio $\Theta = 1400$. Consideriamo una nuova immagine di 256×256 pixels, da ora in poi Δ, che ha le stesse condizioni iniziali delle facce del nostro database iniziale ma che ha una differente luminosità rispetto alle facce presenti nel database. Possiamo vederla in figura 3.9. A questo punto

Figura 3.9: Nuova immagine Δ

procedendo come per le immagini iniziali proiettiamo Δ nello spazio delle autofacce.

$$\forall 1 \leq k \leq 46 \quad \delta_k = u_k^T(\Delta - \Psi) \tag{3.3}$$

In questo caso abbiamo sia sottratto la faccia media Ψ a Δ sia calcolato i coefficienti di proiezione in un'unica equazione 3.3. Così facendo abbiamo trovato le coordinate di $\Delta - \Psi$ nel nuovo spazio. Sia

$$\Omega_\Delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \cdot \\ \cdot \\ \delta_{46} \end{pmatrix}$$

il vettore che descrive il peso di ogni autofaccia per Δ. Ora possiamo calcolare la distanza tra Δ e lo spazio delle autofacce.
Sia $\Omega^T_{\Phi_j}$ il vettore riga dei coefficienti di proiezione di Φ_j, la j-esima immagine privata della faccia media del nostro database. Consideriamo la distanza euclidea tra le proiezioni di $\Delta - \Psi$ e Φ_j

$$\epsilon_j^2 = || \ \Omega_\Delta - \Omega_{\Phi_j} \ ||^2$$

Calcoliamo così la distanza $\forall j \quad 1 \leq j \leq 70$ cioè rispetto ad ogni faccia del database iniziale. Sia

$$\epsilon = \min_{1 \leq j \leq 70} (\epsilon_j^2)$$

A questo punto ci troviamo davanti a due possibili esiti:

1. $\epsilon > \Theta$
2. $\epsilon \leq \Theta$

(**1**) In questo caso si ha che la distanza euclidea tra la proiezione della nuova faccia e una qualsiasi faccia del database iniziale proiettata nello spazio delle autofacce è maggiore del valore di soglia. Questo indica che la nuova faccia non viene riconosciuta come presente nel database.

(**2**) In questo caso si ha successo nel riconoscimento facciale. La distanza infatti è minore o uguale al valore di soglia e questo indica che il soggetto è stato riconosciuto. Il passo successivo è quindi riconoscere quale sia il soggetto a cui la faccia corrisponde. Per far questo consideriamo la quantità studiata precedentemente ovvero ϵ. Per come è stata definita abbiamo che $\epsilon = \min_{1 \leq j \leq 70} (\epsilon_j^2)$. Consideriamo quindi j. Questo ci indicherà il soggetto a cui corrisponde la nuova immagine. Avremo infatti che il volto corrispondente al vettore Γ_j è quello del soggetto riconosciuto. Si noti che a seconda della soglia e delle facce che eventualmente sono presenti nella classe del soggetto, cambia molto la reazione del programma di riconoscimento rispetto all'inserimento di una nuova immagine da riconoscere. Se la soglia è molto piccola il riconoscimento facciale avviene solo in presenza di espressioni e pose molto simili a quelle delle facce iniziali. Se invece la soglia è "giusta" il riconoscimento

avviene correttamente anche in presenza di espressioni facciali differenti da quelle iniziali.

Nel nostro caso, in presenza di una soglia $\Theta = 1400$ il programma non riconosce Δ come presente nel database. Il valore ϵ è equivalente a 1141.8 che è strettamente maggiore di 1400 quindi il soggetto non è presente nel database.

Consideriamo ora tre foto Δ_1, Δ_2 e Δ_3 rappresentanti rispettivamente un soggetto presente nel database, un soggetto non presente nel database e un albero. Poniamo due diverse soglie: $\Theta_1 = 1400$ sotto al quale l'immagine viene riconosciuta come faccia presente del database e $\Theta_2 = 1800$ sopra alla quale l'immagine non viene più considerata come rappresentante un volto. Calcoliamo, come visto in precedenza, la distanza tra le tre immagini e le facce del database inziale proiettate nello spazio delle autofacce. Indichiamo con $\epsilon_{\Delta_i} \quad \forall 1 \leq i \leq 3$ la distanza minima tra l'i-esima immagine da riconoscere e le proiezioni delle facce del database iniziale. Otteniamo:

- $\epsilon_{\Delta_1} = 1141.8$
- $\epsilon_{\Delta_2} = 1661.2$
- $\epsilon_{\Delta_3} = 2097.3$

Figura 3.10: Soggetto presente nel database, Soggetto non presente nel database, Oggetto sconosciuto

Questi valori ci indicano che:

Δ_1 è un soggetto presente nel database. In particolare, passando alla determinazione del soggetto a cui corrisponde si trova che $j = 31$ e quindi che abbiamo riconosciuto nella foto Γ_{31} come si può notare nella figura 3.11 che riporta le prime 64 facce rispetto alle 70 presenti nel database iniziale.
Δ_2 è un soggetto non presente nel database.
Δ_3 non rappresenta un volto.

A questo punto è possibile anche compiere l'aggiornamento del training set.
Ogni volta che viene riconosciuto un soggetto come non presente nel database, viene inserito nel database iniziale e rigenerato lo spazio delle autofacce. Ogni volta che viene riconosciuto un soggetto come presente nel database viene aggiunto allo spazio nella classe di quel soggetto. Questo permette un continuo aggiornamento del database e dell'algoritmo. Se ad esempio, rispetto al nostro database, trovassimo una faccia corrispondente ad un soggetto già presente nel database, la classe di quel soggetto avrebbe due rappresentanti e non più uno come all'inizio.

3.3 Sintesi del procedimento delle autofacce

Riassumendo i passaggi principale della procedura per la generazione delle autofacce sono:

1. Acquisire un database iniziale di M immagini che abbiano delle condizioni prestabilite quali: stessa dimensione, stesse condizioni di luce,

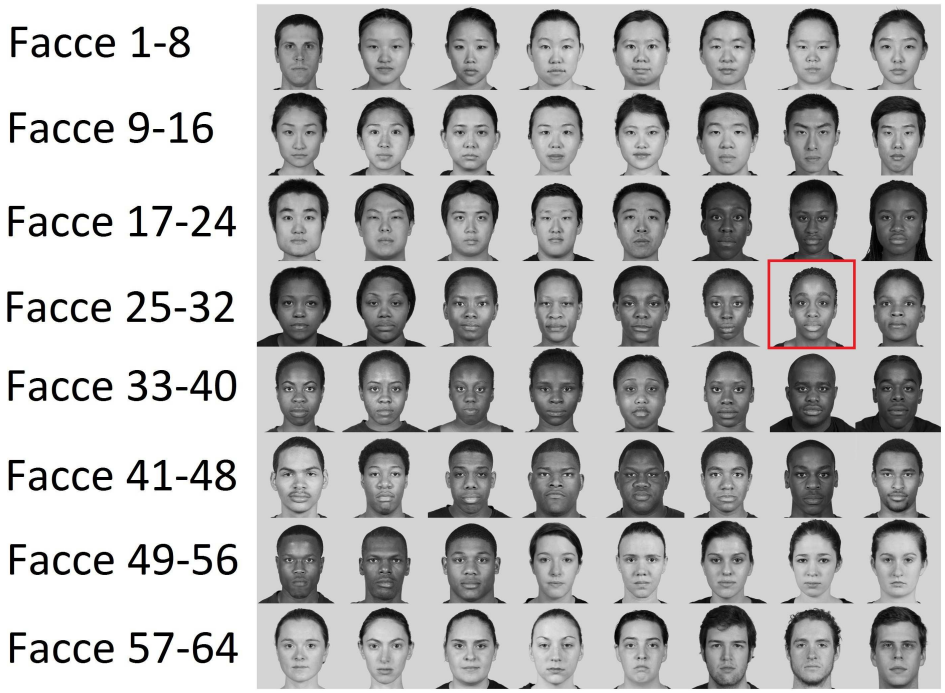

Figura 3.11: Soggetto presente nel database corrispondente all'immagine Γ_{31}

stesse inquadrature etc.

2. Trasformare le immagini in vettori $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}$
3. Calcolare la faccia media $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$ e sottrarla ad ogni immagine ottenendo nuovi vettori: $\{\Phi_1, \Phi_2, \cdots, \Phi_M\}$
4. Calcolare $A = [\Phi_1, \Phi_2, \cdots, \Phi_M]$ e la matrice di covarianza $C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T$
5. Calcolare M autovettori di C
6. Riordinare gli autovettori in ordine decrescente in base ai loro autovalori associati
7. Scegliere un valore η sotto al quale gli autovalori ed i loro autovettori associati vengano eliminati
8. Proiettare le facce del database iniziale nello spazio generato dagli autovettori rimasti

La procedura per il riconoscimento facciale, cioè dato un database iniziale calcolare se un immagine rappresentante un volto è presente nel database, prevede invece questi passaggi principali:

1. Scegliere una soglia Θ sono al quale il soggetto è considerato non riconosciuto.
2. Trasformare la nuova immagine Δ in vettore
3. Sottrarre la faccia media alla nuova immagine
4. Proiettare Δ nello spazio
5. Calcolare la minima distanza euclidea ϵ tra Δ e $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}$
6. Verificare se $\epsilon < \Theta$
   - Se $\epsilon < \Theta$ la faccia è presente nel database. Si determina a quale soggetto corrisponde la faccia individuando la j-esima faccia, Γ_j, che minimizza la distanza tra la nuova immagine e lo spazio delle autofacce cioè j t.c. $\epsilon = \min_{1\leq j\leq M}(\epsilon_j^2)$. In via opzionale si aggiunge la nuova immagine alla classe del soggetto.
   - Altrimenti il soggetto non è presente nel database. In questo caso il nuovo soggetto viene inserito nel database iniziale come nuova classe

   Si rigenera lo spazio delle autofacce.

3.4 Imparare a riconoscere le facce

L'importanza di questa tecnica, oltre ad essere stata una rivoluzione per quanto riguarda la storia del riconoscimento facciale, è il tipo di approccio con cui affronta il problema. Quando abbiamo costruito lo spazio delle facce lo abbiamo fatto a partire dallo stesso database iniziale. Potremmo quindi dire che informazioni necessarie per la soluzione del problema sono presenti nei dati stessi.
Costruendo la matrice A abbiamo parlato di *training set*. In italiano potremmo definirlo "insieme di addestramento". Anche se il nostro scopo è quello di capire se una faccia si trova all'interno di esso infatti, sarà proprio questo insieme a darci le informazioni necessarie per risolvere il problema. Non appena identifichiamo un'immagine come faccia la aggiungiamo al database iniziale. Da un lato quindi possiamo estrarre le caratteristiche che ci permettono di sviluppare l'algoritmo; dall'altro il training set viene aggiornato dalle nuove facce che aggiungiamo. Siamo di fronte ad un tipo di problema che viene definito "the learning problem". Riporto una descrizione di questo tipo di problema che mi sembra esplicativa.[1]
"There is a target to be learned. It is unknown to us. We have a set of examples generated by the target. The learning algorithm uses these examples to look for a hypothesis that approximates the target."
La differenza dai problemi a cui siamo abituati è che qui non esiste una soluzione analitica al problema che prescinde dai dati ma abbiamo una sorta di apprendimento automatico da parte dell'algoritmo usato per arrivare alla soluzione.
Si può quindi sintetizzare dicendo che vorremmo conoscere come si comporta un'applicazione $f : I \to O$ dove I è l'insieme dei dati in input e O è l'insieme dei dati in ouput. Il training set è quindi un insieme di coppie (input,output) che ci permette, in diversi modi, di ricostruire la funzione. Vi sono tre principali tipi di apprendimento automatico cioè tre modi di sfruttare il training set per il nostro scopo:

- Apprendimento supervisionato
  Questo è il tipo di apprendimento più comune. Il training set contiene coppie (input,output)dove entrambi sono ben definiti. Ad un input corrisponde un preciso output. Se ad esempio vogliamo conoscere una funzione f e conosciamo dei valori che essa assume, l'approssimazione della funzione grazie a questi è un tipo di apprendimento supervisionato

- Apprendimento con rinforzo
  In questo tipo di apprendimento i dati del training set non forniscono un output esatto. A differenza dell'apprendimento supervisionato a un

dato in input del training set corrisponde un output con "grado" per questo output. In base alla scelta dell'input il grado dell'output indica quanto la scelta di quel preciso input sia stata corretta rispetto al problema che si tenta di risolvere. I dati del training set però non dicono quale sarebbe stato il grado degli altri output per questo tipo di input. In astratto questo apprendimento può sembrare difficile da capire. Un esempio chiarificante per questo tipo di apprendimento è quello di un algoritmo che implementa un giocatore virtuale di un gioco come ad esempio gli scacchi. A priori, data una situazione iniziale di partita, è difficile predire quale sia la miglior mossa che il giocatore può compiere. In questo frangente usare l'apprendimento con rinforzo significa porre volta per volta il giocatore di fronte a diverse situazioni iniziali ovvero dare in input vari dati e riportare quanto questi siano corrette restituendo output e grado di output rispetto alla riuscita della partita. In questo modo il giocatore virtuale impara a "giocare" "giocando".

- Apprendimento non supervisionato
  I dati nel training set in questo caso sono costituiti solo dagli input in quanto non viene data alcuna informazione rispetto agli output. Con questo tipo di apprendimento si tenta di riconoscere schemi e strutture insite nei dati al fine di classificare i dati in delle categorie a cui non è necessario dare un nome. Un esempio pratico di apprendimento non supervisionato per quanto riguarda la vita quotidiana è quello dell'apprendimento di una nuova lingua. Un modo per studiare nuove lingue è studiarne la grammatica, i termini, la pronuncia etc. Un altro modo è quello di andare nel paese dove quella lingua viene parlata e ascoltare chi la parla direttamente. In questo modo si può imparare a conoscere intuendo in quali contesti vengono usati certi termini piuttosto che altri, conoscendo i modi di dire e apprendendo tutte quelle cose che solamente con lo studio nozionistico non sarebbe possibile imparare.
  L'apprendimento non supervisionato corrisponde in pratica al secondo modo in cui si può imparare una lingua.

L'apprendimento automatico e il concetto di "learning problem" fanno parte della branca della matematica chiamata **Machine Learning**.
La tecnica delle autofacce sfrutta il metodo dell'apprendimento non supervisionato e in parte anche l'apprendimento con rinforzo. Dal database iniziale, il nostro training set che non dà esempi di output, possiamo estrarre le caratteristiche utili a costruire l'algoritmo. Questa è l'apprendimento non supervisionato. Alla fine dell'algoritmo il training set si rinnova in modo da "imparare" nuove facce e la sua "capacità" di riconoscerle. Questo corrisponde all'apprendimento con rinforzo.

Conclusione

In questa tesi si è visto come sia semplice la costruzione dello spazio delle facce e il successivo utilizzo di essa per il riconoscimento facciale limitato ad immagini che riportano volti nella stessa condizione di posa e rotazione della testa. Abbiamo ripercorso la verifica fatta in laboratorio di questa tecnica spiegando passaggio per passaggio la procedura.

Appendice A

Codice

Di seguito viene riportato il codice utilizzato per la costruzione delle immagini presenti nella tesi. Il codice è stato implementato a partire da quello proposto da Michael Scheinfeild che si può trovare sul sito MathWorks.com, in particolare all'indirizzo
it.mathworks.com/matlabcentral/fileexchange/45915-eigenfaces-algorithm
Il codice prevede che le immagini siano già in bianco e nero e che la loro grandezza sia 256×256. Nel caso si considerino immagini di diversa grandezza (sempre immagini quadrate) è sufficiente cambiare il dato corrispondente a siz. La cartella da cui vengono caricate le immagini in origine è C:\University\ Thesis\MR2 Face Database. Il database su cui è stato mandato, come detto in precedenza, è parte del MR2 face database.

```
%%
%Le immagini sono state tutte ridimensionate in 256x256 e trasformate in
scala di grigi
%Hanno inoltre stesso sfondo e stessa angolazione della testa.
%Con la scritta DISPLAY ci si riferisce a linee di codice che hanno il solo
%scopo di visualizzare le immagini, non sono utili nell'algoritmo per
%la creazione delle eigenfaces, in particolare le immagini si riferiscono
%solo al numero quadrato più vicino al numero di facce nel database
%in modo da poter visualizzare delle
%matrici quadrate


d=dir('C:\University\Thesis\MR2 Face Database\');
siz=256;
d(1)=[]
d(1)=[]
n=length(d);

%Ciclo che carica e salva nel database db le immagini di una cartella
for i=1:n;
    a=strcat('C:\University\Thesis\MR2 Face Database\',d(i).name);
    b=imread(a);
    b=im2single(b);
    db.nome{i}=d(i).name;
    db.dato{i}=b;
end


%Matrice con alcune facce del database iniziale          %BEGIN DISPLAY

q=1;
s=fix(sqrt(n-2));
MF=zeros(s-1,s-1);
for i=0:s-1;
    for j=0:s-1;
     MF((siz*i+1):(siz*i+siz),(siz*j+1):(siz*j+siz))=db.dato{q};
     q=q+1;
    end
end
figure(1),imshow(MF,'initialmagnification','fit');;title('Database
Originale')
imwrite(MF,'Database originale.jpg');                   %END DISPLAY

%Analisi delle componenti principali
%PCA

%Faccia media
fm=zeros(siz);
for i=1:n;
    db.dato{i}=im2single(db.dato{i});
    fm = fm + (1/(n-2))*db.dato{i};
end
figure(2),imshow(fm,'initialmagnification','fit');;title('Faccia Media')
%BEGIN DISPLAY
```

```
imwrite(fm,'Faccia media.jpg');
%END DISPLAY


%Differenza tra le facce originali e la faccia media
%Facce normalizzate
for i=1:n;
    db.datomedia{i}=db.dato{i}-fm;
    imwrite(db.datomedia{i},'Faccia normalizzata.jpg');    %DISPLAY
end

FN=zeros(s-1,s-1);                                          %BEGIN DISPLAY
w=1;
for k=0:s-1;
    for j=0:s-1;
     FN((siz*k+1):(siz*k+siz),(siz*j+1):(siz*j+siz))=db.datomedia{w};
     w=w+1;
    end
end


figure(3),imshow(FN,'initialmagnification','fit');;title('Matrice delle facce
normalizzate')
imwrite(FN,'Matrice delle facce normalizzate.jpg');
                                                    %END DISPLAY

%Trasformazione delle facce in vettori colonna di lunghezza siz^2
A=zeros(siz^2,n);
for i=1:n;
    A(:,i)=db.datomedia{i}(:);
end
size(A)
%Matrice utile a calcolare gli autovettori della matrice delle covarianze
C=A'*A;
xlswrite('Matrice_A^TA',C)

%Autovettori e autovalori della matrice delle
%covarianze
[Avet,Aval]=eig(C);
Agrande=A*Avet;
xlswrite('Autovettori_di_A^TA',Avet)
size(Agrande)
%Autofacce
autofacce=[];                                           %BEGIN DISPLAY
for i=1:n
    c=Agrande(:,i);
    autofacce{i}=reshape(c,siz,siz);                    %END DISPLAY
end
size(autofacce)

%Riordine degli autovalori
x=diag(Aval);
[xc,xci]=sort(x,'descend')
xlswrite('Autovalori_A^TA',xc)
```

```
%Matrice delle autofacce senza scrematura     %BEGIN DISPLAY
mautofacce=zeros(s-1,s-1);
l=1;
for i=0:s-1;
    for j=0:s-1;
     mautofacce((siz*i+1):(siz*i+siz),(siz*j+1):(siz*j+siz))=
autofacce{xci(l)};
     l=l+1;
    end
end
size(mautofacce)
figure(4), imshow(mautofacce,'initialmagnification','fit');;title('Autofacce
senza riduzione')
imwrite(mautofacce,'Autofacce senza riduzione.jpg');
                                                %END DISPLAY

%%Numero delle autofacce che non apportano contributi utili
sommaav = sum(xc);
z = 0;
for i = 1:n
z = z + xc(i);
tv = z/sommaav;
if tv > 0.95
k95 = i;
break
end ;
end
nc=i


%Matrice delle autofacce utili      %BEGIN DISPLAY
y=1;
s_2=fix(sqrt(nc-2));
spfacce=[];
for i=0:s_2-1;
    for j=0:s_2-1;
     spfacce((siz*i+1):(siz*i+siz),(siz*j+1):(siz*j+siz))= autofacce{xci(y)};
     y=y+1;
    end
end
figure(5), imshow(spfacce,'initialmagnification','fit');;title('Autofacce');
imwrite(spfacce,'Autofacce.jpg');    %END DISPLAY


%Calcolo del peso delle facce originali nello spazio delle facce(autofacce)
for i=1:n;
    for j=1:nc;
        pe(i,j)=sum(A(:,i).*autofacce{xci(j)}(:));
    end
end
xlswrite('Coefficienti di proiezione',pe)
for j=1:nc;
 vetauto(:,j)=reshape(autofacce{xci(j)},siz*siz,1);
end

pe1=zeros(siz*siz,1)                                %BEGIN DISPLAY
```

```
for j=1:nc;
    pe1(:,1)=pe(1,j)*vetauto(:,j);
end
PE1=reshape(pe1,siz,siz)
figure
imshow(PE1,'initialmagnification','fit')
imwrite(PE1,'Proiezione del primo soggetto.jpg')  %END DISPLAY


%Riconoscimento facciale
f=dir('C:\University\Thesis\Test Database\');
f(1)=[]
f(1)=[]
m=length(f);
s=1;
for i=1:m
    a=strcat('C:\University\Thesis\Test Database\',f(i).name);
    b=imread(a);
    b=im2single(b);
    dbp.nome{i}=f(i).name;
    dbp.dato{i}=b;
    newfaccia=b(:)-fm(:);
    for i=1:nc;
       pfaccia(i)=sum(newfaccia.*autofacce{xci(i)}(:));
    end
%Calcolo della distanza euclidea
    for i=1:n;
        sc=0;
        for (j=1:nc)
          sc=sc+(pfaccia(j)-pe(i,j)).^2;
        end
        dp(i)=sqrt(sc);
        vet(i)=dp(i);
    end
    mind=find(vet==min(vet)) %trova l'indice della faccia che minimizza la
distanza
    z=min(vet)

           if z<1400;

figure(n+5+s),imshow(b,'initialmagnification','fit');;title('Soggetto con
corrispondenza nel database')
            imwrite(b,'Soggetto con corrispondenza nel database.jpg');
           s=s+1;
           elseif z<1800;

figure(n+5+s),imshow(b,'initialmagnification','fit');;title('Soggetto senza
corrispondenza nel database');
            imwrite(b,'Soggetto senza corrispondenza nel database.jpg');
            k=k+1;
           s=s+1;
           else;

figure(n+5+s),imshow(b,'initialmagnification','fit');;title('Oggetto
sconosciuto');
            imwrite(b,'Oggetto sconosciuto.jpg');
```

Appendice B

Autovalori di $A^T A$

31749.77179
14553.17369
6850.326198
5515.631419
3859.547448
3447.076782
2953.126297
2107.076921
1878.307464
1595.159827
1377.133281
1322.99345
1278.980185
1202.032555
1171.271412
1030.948513
983.0510463
913.0924779
859.810696
811.6996537
729.3379761
683.0016553
669.5901846
652.3033589
647.1079749
587.6269435

Continua nella prossima pagina

561.2599254
522.290991
503.0403741
493.0723634
467.944998
454.6703918
442.9205139
424.9442627
402.8962252
398.3464663
385.8948387
375.710714
339.0807339
336.886422
332.7580573
318.8316932
316.4553485
301.5345253
300.3193523
283.278646
276.0691122
274.4133907
257.939447
253.0459257
245.9309642
242.321986
236.9875403
226.0877817
222.666564
217.5744613
211.1324236
202.8192593
194.4256959
192.6890091
186.8503096
181.2920584
176.2517739
175.0466873

Continua nella prossima pagina

160.5617117
159.5674533
151.3613682
145.3942189
138.9245183
135.3407718

In grassetto vengono riportati i primi 64 autovalori della matrice $A^T A$. Rappresentano gli autovalori i cui autovettori corrispondenti sono utili alla riduzione della dimensione dello spazio iniziale.

Appendice C

Matrice dei Coefficienti di proiezione

	1	2	3	4	5	6	7	8	9	10
1	2395.29	2216.73	-938.528	6.46964	218.304	247.1561	-837.565	-102.359	151.2517	79.37951
2	-2372.72	-1584.05	214.5712	-351.28	-670.226	44.43591	-34.699	-100.418	51.80257	-116.775
3	-1172.3	-626.955	234.7273	-1006.39	-54.9036	351.5462	448.551	-42.1413	175.5897	-40.0815
4	-3339.26	932.2895	933.9018	415.2349	-389.203	-445.825	127.5285	112.963	-4.8094	56.476
5	-3291.09	-442.679	-277.173	974.4555	-209.833	-28.6093	343.0938	-147.998	61.90118	-66.9682
6	-2870.41	-21.0045	-388.339	885.5537	-54.8322	61.54647	438.6249	-155.387	-5.86758	-168.624
7	-3232.69	147.3714	717.4708	-370.848	-39.0412	148.2847	448.2008	15.55024	100.3351	-159.28
8	-2637.29	-117.286	-137.359	136.976	175.0485	117.1304	430.0689	-6.55744	-55.1234	-192.522
9	-2154.12	-2104.11	125.9685	-891.07	-77.7089	87.1511	-325.848	-151.318	-187.967	-139.296
10	-1261.84	197.4183	-55.4605	663.6491	1154.215	-212.398	12.60783	299.5226	-190.335	-141.267
11	-3021.5	-1467.06	208.651	29.67477	-94.1562	-199.456	85.29428	-31.7797	-239.674	1.202522
12	-3829.36	579.972	-68.6006	724.9727	-232.121	-92.2072	408.8947	-135.629	-116.174	-59.0553
13	-3038.49	-481.515	-227.485	-483.135	613.5701	-151.12	98.80636	-296.974	320.7198	20.76204
14	-2705.76	-3592.12	-95.2625	382.7781	51.6912	485.925	-434.548	155.8013	-207.601	402.0438
15	-281.494	-2109.17	-472.028	-440.799	-445.003	-291.188	-32.6508	-238.88	61.48216	185.688
16	-958.547	-706.249	-1077.54	-213.493	477.8702	-14.9579	36.91732	-541.39	321.1307	-111.902
17	-2976.08	-2956.81	-427.611	532.3588	-251.141	351.8983	-230.169	-224.38	-290.406	232.0487
18	222.3517	-819.188	-1129.34	1542.427	620.8015	181.6024	-160.199	36.03058	-276.346	117.2863
19	-1436.46	-2888.38	-163.19	-66.3808	167.861	-42.7482	-399.802	-27.3006	-269.225	-14.3346
20	1087.84	1270.529	139.3975	-81.3687	1460.006	-218.179	-261.357	295.334	145.0349	47.04716
21	-128.73	-1431.46	-1220.07	1021.13	75.48253	89.63744	176.9059	-186.646	-234.485	-63.6642
22	7199.425	-309.874	-1195.86	-441.008	-480.619	-720.791	135.1267	236.8053	64.52351	-214.277
23	5749.834	-347.414	-88.9256	-2045.25	-296.742	29.40503	516.7402	143.5324	-152.31	219.438
24	6655.623	-5049.86	1002.733	1111.932	-108.069	80.70002	659.6291	236.0053	577.4828	843.2851
25	6271.265	-4662.76	1329.73	897.4705	454.0613	281.6665	506.3561	-387.612	-383.601	-277.971
26	3438.071	-3603.61	322.3383	6.639003	309.6068	-22.1003	515.8952	656.5995	-375.667	-451.824
27	1183.316	-522.91	-978.595	203.1283	-326.686	-169.75	419.5668	-75.3216	192.35	-239.959
28	716.0625	2028.353	-732.576	26.39863	132.9413	775.5782	207.299	-77.848	-129.218	108.326
29	1774.584	-2383.67	-1201.49	86.12748	-661.181	-10.7245	190.2166	-189.589	-63.666	40.46688
30	3121.042	1382.263	-830.897	-536.624	269.9608	276.6535	180.6091	-38.7533	-36.0526	40.02278
31	2235.56	1745.665	168.4719	-489.232	-68.3172	57.0684	184.3821	-33.5619	-112.845	164.2374
32	2647.193	1162.464	-1127.74	190.6875	137.7017	380.81	256.8554	26.19929	219.0278	2.337592
33	2887.307	577.9544	-183.135	-1230.5	-102.664	-190.412	492.3641	-42.4813	-93.3031	78.18258
34	1873.433	-1063.16	1672.085	-735.558	-417.728	-509.153	-124.514	-402.189	-356.701	195.8197
35	5631.967	2232.712	441.7773	-174.611	59.45884	767.2718	-462.211	-325.488	-591.258	212.4822
36	2203.885	-2708.58	-879.669	-1137.9	-827.328	441.783	-311.504	28.98572	56.28898	-321.249
37	2155.216	-220.855	-1022.57	539.5332	96.3331	-503.823	-38.1164	-25.618	-258.521	-12.8011
38	2288.847	566.264	-687.054	-781.188	-429.008	882.4162	686.5336	102.8494	-102.282	100.5257
39	9725.962	4.756736	1914.131	921.0687	-344.826	8.958979	-181.091	-376.849	478.5959	-357.152
40	10617.37	324.7049	1761.548	640.0298	-28.9993	330.2529	-175.636	-581.53	409.9712	-258.832
41	-3234.9	565.0486	491.5321	271.2816	-771.174	-434.037	-17.4922	-16.4128	98.29852	117.9226
42	1583.147	756.3567	-707.312	-361.599	180.63	-599.024	-205.294	-292.981	97.2777	113.2733
43	5819.438	1421.615	1590.715	75.78354	274.7994	268.3823	-146.464	-246.753	-345.608	182.9823
44	3813.963	599.1971	-348.323	415.1694	-227.086	-232.873	60.66416	585.1949	272.7048	162.0152
45	7574.906	582.0416	384.935	553.5845	182.7685	411.8602	-197.563	551.3765	110.8996	-189.24
46	450.2402	-126.278	-102.614	-763.529	-34.5495	-685.478	-24.816	-93.0218	191.5141	118.5309
47	6452.329	293.5107	-1238.54	548.2156	-786	-186.731	-328.954	251.8856	-32.3211	-79.0556
48	2582.344	1230.051	-193.327	-344.324	147.6145	-592.425	-264.413	136.3088	20.36815	-49.9333
49	6120.076	1397.604	633.4361	-369.388	116.9409	104.4014	-22.1554	-41.8574	-260.645	249.0886
50	5324.784	1745.695	-1105.58	158.7555	-500.77	433.5722	-626.216	247.6867	-2.17586	-60.6816
51	2409.918	-608.817	-376.56	464.5854	179.2872	-112.066	609.66	338.1068	158.1997	214.606
52	-1759.94	199.1424	434.5515	-113.234	307.3507	-872.464	-143.767	-108.385	103.8571	95.93586
53	-3980.8	1603.492	1054.363	-407.236	-896.078	227.2349	50.38303	105.8044	130.0518	139.8413
54	-1220.66	-738.791	1079.706	-924.437	42.6479	-483.679	-148.642	180.7696	149.9231	-45.879

	1	2	3	4	5	6	7	8	9	10
55	-3934.35	-242.535	1247	-879.042	278.3751	1169.041	-48.8541	424.7059	92.77799	-180.201
56	-5429.28	-1529.89	789.4126	-45.9752	-651.508	590.4297	-551.918	110.3208	141.2426	-199.609
57	-4559.52	798.1333	528.9432	199.2634	-4.27082	928.6548	282.4425	55.04001	210.8124	-35.1465
58	-4381.5	1162.426	398.4307	27.85151	-302.821	154.3731	380.2802	-113.255	80.15	-30.2186
59	-1833.89	1315.83	1600.592	440.1931	315.0379	-486.002	149.2206	496.5216	-81.8942	-9.35538
60	-4512.73	2232.239	1252.724	602.5711	-433.838	30.70672	227.712	283.0503	-256.696	32.35449
61	-3078.58	556.3946	-142.613	501.3323	143.9764	-477.735	106.0071	-325.314	-96.5418	-34.5142
62	155.5523	-3451.98	-147.352	-662.465	301.3172	2.923216	-670.465	337.2813	285.6038	-123.451
63	-2278.86	-315.676	-286.309	794.3627	-234.442	-57.1786	75.32453	-112.453	253.4307	-181.177
64	-1355.72	-2664.66	-530.123	232.808	-351.688	357.3069	-333.54	-16.4115	166.8068	100.9323
65	-2413.89	2307.29	487.0065	1429.161	-971.565	1.51797	-716.118	179.2968	35.73415	107.3025
66	-2637.59	1573.68	-333.258	-95.8547	754.9561	691.2537	222.072	-330.039	343.1916	90.07352
67	-1629.39	-2459.06	877.8062	-789.448	1083.517	-194.053	-496.245	-50.0785	131.689	1.578156
68	-3178.93	-2185.54	-164.306	-34.6251	248.0944	347.2065	-627.998	45.35648	222.2826	244.9049
69	-2167.37	1068.605	85.87703	-162.606	-18.711	-175.651	163.6635	-164.37	164.5793	65.64633
70	627.6897	1135.383	-904.601	48.58785	567.9628	540.9341	203.5462	-73.1203	252.4016	89.43831

	11	12	13	14	15	16	17	18	19	20
1	-48.4194	100.9802	5.139842	-145.817	104.9195	65.48991	-230.971	-333.447	-40.4618	100.0001
2	-100.055	-13.6039	-33.7299	103.0542	-210.088	18.88152	21.34381	-125.785	106.108	3.444416
3	-273.536	21.92917	162.8467	91.03581	-207.087	-78.7571	-24.1576	-102.614	83.07642	-96.1762
4	-262.055	68.75777	257.6803	-34.1561	45.52115	-205.821	70.04863	-29.6666	2.407545	212.1872
5	18.03874	-68.9948	81.52612	-143.977	1.399099	-168.36	82.67562	27.21295	28.95014	109.8524
6	-59.3884	-4.285	-29.1962	-88.0206	-4.6294	-146.961	133.786	68.7841	22.56814	102.8073
7	24.74194	249.7951	-71.397	29.71745	70.96308	-106.817	-88.5371	-74.4624	-55.6815	11.81719
8	-234.022	240.0699	-73.464	-262.345	-55.3912	76.17168	-14.7543	-112.499	48.46188	92.31241
9	-204.748	64.18182	-443.355	-50.118	-83.734	-23.5364	-24.0351	6.633224	28.12969	8.140552
10	-407.642	-34.0264	-170.331	-141.869	-195.614	0.73598	10.19248	-120.134	158.4786	-39.915
11	-91.892	264.1625	-165.345	85.81848	-160.782	4.56904	-11.5134	-182.686	166.7892	107.3282
12	-136.325	182.296	-34.5395	-112.631	80.87986	-162.813	80.18327	58.84382	45.82883	127.3425
13	-84.375	276.4122	70.20834	-202.931	27.93356	-43.6749	-32.2296	33.6192	159.8872	-232.477
14	-538.969	-34.1702	310.1165	-54.838	350.5117	142.4888	154.2523	54.39595	-96.2094	-55.9787
15	-163.381	198.9175	185.456	112.9181	43.2959	-1.51883	25.51583	-20.8651	-50.3894	-16.0484
16	110.2112	-48.7156	87.91024	-245.39	-10.782	-181.783	1.426038	7.771018	149.9173	76.58726
17	-259.605	96.08212	22.69424	288.033	132.6447	-146.036	-249.512	133.2705	13.52408	141.2914
18	-167.346	-58.9283	26.03001	-11.1055	46.2742	-116.004	-9.60462	-175.232	-78.9972	-164.533
19	-94.4516	68.52777	-283.933	226.5094	32.72874	49.52068	-234.518	121.4859	71.27794	68.69852
20	-108.398	101.582	-155.46	276.5322	14.23794	-54.0741	84.21539	-177.909	-41.4086	70.95179
21	-51.4908	-124.685	-40.0177	-81.8301	30.01272	-232.431	14.86637	77.47065	-51.5369	86.8443
22	-230.213	-281.512	13.12779	-75.5835	111.883	-132.013	23.9917	-13.375	82.81651	68.52706
23	-175.325	-38.6384	144.0899	-53.6486	43.73927	-338.416	-133.804	49.01718	-64.0836	-36.9731
24	44.00062	-190.02	-408.405	-225.454	-266.918	-68.9556	-72.175	-136.743	-60.7012	52.50672
25	85.46655	-96.4659	251.7449	40.89801	267.1342	277.2492	-122.319	-190.453	190.7324	-41.3462
26	178.983	-169.015	150.1129	-96.4804	125.3383	-72.4703	-94.0164	-129.186	-17.8542	-71.9387
27	88.62852	-86.0022	15.45615	-78.4862	57.47677	8.386563	198.5675	-111.665	81.23331	78.45561
28	-8.3791	-36.9001	-41.793	245.1393	-51.0895	-138.411	248.6167	-56.7343	103.5634	-42.8801
29	83.49097	80.71585	82.53869	135.232	-12.4998	-72.6772	48.63445	-220.77	53.08432	94.94002
30	-188.769	-165.155	-165.199	69.5162	93.34719	-75.411	141.1043	-35.67	5.560741	-52.2662
31	-113.945	-431.608	13.33844	110.0329	83.31268	-65.7562	-19.6215	53.16549	105.9849	-5.04501
32	27.84584	-66.2599	-96.2354	314.8698	125.9138	48.52477	93.97668	-140.691	65.18399	-135.535
33	-136.7	-102.703	-155.582	-102.914	142.5591	-9.86479	-133.11	44.73664	40.31337	-41.2438
34	278.7861	40.58536	-165.361	3.453492	103.1964	10.55128	141.9774	-213.605	-29.515	-52.1197
35	157.1923	84.85533	55.91187	-283.776	55.0812	-278.905	-57.9429	-80.0657	27.89739	33.61506
36	45.42563	39.95789	-365.459	-52.2627	158.8093	69.56756	159.201	-29.1713	10.1514	93.49747
37	-44.2034	31.58641	-239.296	58.35763	155.4919	-169.488	81.86687	58.49207	-56.8431	-242.837
38	90.67455	266.5098	-55.0839	65.46125	74.35632	-18.9759	-22.5993	74.15818	177.7764	-120.149
39	-157.874	68.43561	-44.1755	8.531225	55.88478	-37.0904	-17.5188	123.2013	-17.483	-32.9385
40	-413.826	145.7244	-110.906	130.3985	43.89624	-127.415	16.50594	58.79857	-66.2656	-96.1833
41	26.22117	-73.4707	94.43219	59.99078	150.5269	6.958007	3.368632	-182.964	134.4077	-166.781
42	-148.704	-103.229	115.5818	-276.236	57.50371	63.54066	85.67283	-25.1371	164.7938	-24.346
43	7.737062	24.49892	114.4595	-6.42242	-26.4704	143.6353	304.406	47.89176	165.5613	204.0967
44	-166.387	30.83442	136.6387	93.32315	103.1201	130.8249	1.014294	79.30882	157.3878	98.17934
45	41.96754	338.6116	178.3552	51.88771	-48.6823	-188.811	-84.6368	38.6284	90.78944	21.94512
46	-97.0224	75.53443	53.59507	-55.0639	89.41659	74.3711	-11.5024	-105.455	93.59179	-32.721
47	-13.5704	145.1155	-91.0841	-141.065	106.8741	-29.3576	21.66793	-11.2856	-84.5843	-123.385
48	-128.728	-68.6993	-187.972	-132.963	210.8337	141.658	-164.057	61.27707	87.63259	111.0869
49	110.9413	-204.017	-140.742	-4.81748	43.08903	28.2534	100.3175	84.29066	81.72544	134.7253
50	-85.6193	40.3059	8.670406	-88.3775	-81.476	3.851266	-166.631	-234.305	30.64271	151.2598
51	115.0514	377.2176	2.121262	-132.239	284.1634	99.42366	180.0692	28.59832	-29.8822	63.16268
52	-206.388	-127.483	9.557831	210.4523	60.3761	-81.2027	76.66392	-142.754	22.81495	-38.6568
53	-165.454	-37.5285	113.4008	32.49605	78.6299	-16.5777	-100.764	-59.7306	78.60168	-18.6789
54	83.52482	180.986	131.6384	162.2922	120.4866	-217.926	120.709	-208.539	-88.3784	102.3935

	11	12	13	14	15	16	17	18	19	20
55	-147.682	-65.8859	-75.8531	-222.186	164.6319	-43.8989	215.0853	-107.453	-211.065	14.68997
56	-110.496	-274.195	-145.221	-68.4282	81.96001	-2.5075	131.7465	-60.4364	24.45328	-42.1399
57	-64.0537	-308.133	37.92072	-138.023	47.52981	102.7883	-158.646	13.99446	57.66165	-34.1532
58	-48.4379	109.8329	-83.0485	-105.233	172.6838	-38.4019	-102.164	-87.8013	12.47493	-24.9154
59	152.8181	67.2979	-192.057	108.8993	264.1869	-143.101	-32.3263	54.79328	35.74684	-9.45847
60	-129.296	21.50635	-280.585	29.63519	36.63521	-68.2644	-68.8583	2.314741	283.5034	-7.41871
61	-9.20556	-64.4095	-239.216	-48.2869	236.9702	-75.3424	97.70798	-37.1116	-77.6609	7.860817
62	62.60548	-171.14	47.76541	98.08653	63.92238	-95.858	145.1211	139.3799	216.328	105.2214
63	185.3193	-210.523	-50.4074	167.6167	104.8951	-100.976	-113.307	1.529967	24.45446	66.22997
64	187.8855	48.49226	25.91137	207.8254	82.51083	-123.665	-19.0564	-3.95085	11.03491	22.98335
65	181.168	-137.589	42.22545	-102.988	56.97188	-153.257	2.838056	-78.4355	124.5931	-147.604
66	39.21402	-191.551	3.510981	161.9303	156.8339	-33.0492	-45.5085	-96.423	-58.3099	142.4793
67	229.3445	-134.537	113.3873	-144.133	46.63282	-308.391	-141.163	13.34737	100.369	-34.3603
68	148.8555	278.9951	-78.9328	-210.345	113.507	-26.925	133.7903	128.7499	213.4434	-104.877
69	45.74253	35.32369	-89.64	-81.8872	449.3693	108.7735	-186.691	21.37082	-118.941	117.7432
70	136.365	88.39878	-109.073	143.6421	238.9149	4.639773	-56.5192	-28.7223	162.5927	10.28698

	21	22	23	24	25	26	27	28	29	30
1	-73.8826	70.63471	-19.7474	-72.2292	51.81201	-114.388	75.37155	34.35518	9.103402	4.565197
2	89.21996	-5.8645	-93.4053	12.21165	24.67146	-9.72071	-5.06641	129.7874	-61.9348	33.53526
3	-28.9594	74.33153	-87.3935	178.8573	-21.9902	19.83692	72.80474	-17.1045	59.27131	40.96896
4	25.55164	93.77904	-67.9848	-51.3623	-66.1532	42.62239	-23.6206	-39.3932	-46.3661	-45.5361
5	-6.87628	-135.339	-43.5322	43.36259	14.82537	-121.265	18.38674	6.80706	-1.26828	-64.4733
6	54.66496	16.31084	-40.3479	78.42411	13.4996	-57.1337	12.62224	1.288814	23.72444	-9.40603
7	4.497059	68.24687	-198.27	14.24879	-17.8443	11.16046	-6.18597	57.62241	-39.5934	-23.5263
8	-31.3473	32.09976	27.9536	46.54904	67.90805	-8.13956	-38.4263	-24.4734	70.90106	-31.1146
9	-4.06707	-7.77284	-6.09669	151.7217	-61.0591	-31.2625	27.59447	-53.9356	-3.39721	60.26962
10	-31.7421	68.43675	160.7085	157.319	32.67088	-39.7257	-86.4623	-3.57317	26.7119	-36.2741
11	-68.6189	-262.749	60.0854	39.74091	70.59769	-76.2873	69.0095	12.31027	-27.7844	-14.1692
12	46.098	89.42029	42.90415	-38.8202	-28.772	-32.8046	60.05337	-17.9203	8.163219	6.750681
13	98.68637	198.8045	-109.14	9.574879	23.57602	4.537043	-8.92002	49.57217	0.360747	-62.7563
14	93.10868	47.52604	-28.7607	-23.4785	-23.0571	-60.1076	80.73892	-6.86321	119.2378	-22.0809
15	124.2751	-18.326	43.78169	85.05081	86.91017	-62.0485	53.83302	62.77695	35.63524	28.39448
16	-14.8934	-8.97393	-101.647	-46.3197	-58.7894	113.4171	106.6192	48.50654	77.368	92.44365
17	9.921161	85.01937	-135	-76.8544	-29.7262	88.5807	-101.324	-8.33283	-78.6407	110.9649
18	-154.941	-48.8501	-63.6722	118.6211	-76.9504	79.66588	-114.246	19.15655	-46.3153	-87.0779
19	89.30896	-140.79	-120.113	41.34304	30.13659	74.3606	-44.6805	1.055012	43.85263	-13.0839
20	3.187168	-51.9406	-135.674	-51.5399	-54.4035	3.603384	121.3156	74.94678	-13.4597	-9.75101
21	-76.5117	13.2508	2.787419	-45.2584	93.16194	-30.7157	-61.5336	77.95411	-69.054	33.65168
22	-44.3892	145.9749	129.9722	-16.8962	-26.7369	26.52003	35.86604	101.6426	-73.2779	87.10852
23	-74.4756	-177.579	52.51568	50.08594	-14.3199	47.74436	19.53325	26.25835	-88.809	-120.046
24	128.832	48.7502	6.19784	-73.3224	27.86645	-13.3395	27.51379	-10.4514	-16.835	7.904223
25	31.47292	-37.5858	106.4365	-58.6272	-34.0462	104.1528	50.38803	103.0551	-55.7208	-31.6931
26	-22.9276	22.15614	-149.151	-21.2646	55.26505	-157.598	101.877	-158.062	-3.89365	86.91906
27	85.83295	-56.1371	-61.0203	-44.6039	-4.46301	24.25738	-76.9746	-37.8634	50.81924	-31.6048
28	89.26393	-33.8442	-40.3717	-27.0581	60.57571	-47.7166	8.845424	-30.0973	-9.83265	-48.635
29	-52.1371	1.210333	94.85222	-5.10176	15.76362	102.2392	-85.8877	-75.8753	143.238	-60.3945
30	46.02848	-30.8129	85.14107	-33.9291	-19.8064	111.9422	80.21632	39.703	-49.332	-70.1514
31	-31.8488	5.597669	39.06992	33.53773	9.292217	-58.1791	-91.506	9.214496	168.3065	146.8084
32	13.88977	97.41026	-58.6799	8.904376	-43.9619	-34.3593	-28.044	-49.5837	-64.5273	-13.6142
33	-30.6805	-92.4767	-19.7874	2.954197	-82.4187	-23.135	-34.7859	92.48131	65.31411	39.13045
34	-65.0903	110.819	-72.8208	127.1717	-172.255	-102.115	-80.0523	85.1118	71.50981	26.31826
35	141.4764	-25.4419	20.34006	-24.3933	-169.341	-46.1322	77.80102	-75.1513	-13.1357	-81.2842
36	59.53934	57.9434	-8.24282	-60.3904	-37.0599	38.03425	-31.156	-24.688	-5.81108	-34.34
37	90.92665	-4.18538	-35.7315	-110.244	220.6057	-104.208	-16.1163	123.0515	98.00033	-34.8659
38	24.45233	5.92819	-32.1382	-13.6179	106.8295	-39.3585	98.36911	-41.0213	-8.94067	23.92739
39	49.89229	-63.9032	-1.78268	77.62271	24.13256	70.25371	12.77689	57.35079	60.35151	-95.4331
40	-237.564	-37.1355	-17.1435	-119.638	-3.41201	-114.718	-15.3387	-128.983	5.886811	56.44616
41	62.20208	-40.4373	-31.747	-46.2112	51.62753	-10.0989	-132.401	-127.364	-38.3589	-34.5163
42	-1.28211	-98.9294	-147.599	-12.7375	-110.868	-85.6694	-102.105	-9.49289	-65.1475	39.93768
43	8.079966	46.4276	-58.7532	81.85434	259.6042	47.93571	25.27342	29.89264	-70.3258	64.38127
44	-79.8427	-68.4487	-62.1889	101.0022	-84.2724	25.2133	10.61307	-74.625	3.626439	-18.8822
45	339.4125	-16.5954	106.5227	114.48	-113.661	-99.3192	-157.518	49.4163	-8.36578	91.66924
46	72.66782	-142.755	-59.0671	-53.6063	42.29747	-7.19613	-62.3658	-1.01591	-10.1192	22.30164
47	-3.03788	69.15334	-52.4179	205.2512	106.8697	170.0999	62.084	-35.318	-53.5245	43.58452
48	96.25306	-21.8269	-70.8672	27.69	41.60644	9.712277	20.30883	-60.5289	-24.8061	-75.4887
49	-62.1454	126.4205	-153.508	133.3243	71.23032	-5.97836	-56.5189	-38.7718	-13.8976	-81.1658
50	-39.3623	36.80348	-66.1442	-89.2478	91.918	-10.2991	-40.5345	53.71585	49.23714	-2.51954
51	-173.431	-84.8702	-59.6728	50.93352	-54.6675	43.2745	-29.6472	39.58897	43.93341	45.01162
52	120.1847	8.921951	34.26265	29.03702	-8.17101	56.02297	150.7211	-106.077	-49.3376	53.66675
53	-79.914	-21.6958	50.54412	-31.324	153.0605	-78.3285	-80.9394	49.30016	-39.5954	-66.4758
54	-42.8687	47.57879	77.20187	-6.34498	67.61184	12.08256	-25.2527	46.76652	-62.5346	42.0891

	21	22	23	24	25	26	27	28	29	30
55	20.59121	-165.41	-59.984	-41.1176	52.85006	96.00654	-75.2357	16.39365	35.45832	92.15633
56	93.99537	44.23175	49.24958	28.88462	-56.8593	14.77003	-27.518	-28.9063	-58.4296	-49.3991
57	-55.289	31.38889	-63.0731	75.84485	11.87918	-66.1909	8.268181	128.0044	-121.123	11.78656
58	-49.1704	104.363	24.88057	5.326876	8.671116	82.21243	14.03963	-21.089	32.85836	-62.2938
59	-1.40752	63.2439	-33.4202	-42.8697	-17.4312	35.43941	-31.7851	95.32768	36.71573	-64.7606
60	-75.4964	40.72162	58.36764	-135.15	-77.3862	65.82837	83.68956	-27.5903	79.20542	40.57736
61	23.06335	-56.8702	86.51179	44.58648	3.006839	-48.2409	-11.8243	-20.1379	-124.794	86.85453
62	-82.2209	26.38928	16.38613	-52.5742	-41.8841	-112.829	52.51043	67.21463	50.7374	-129.86
63	39.83683	-93.887	-139.202	98.28787	5.662752	-68.7358	37.07484	-29.829	-18.4092	-45.8135
64	-191.331	73.49879	77.51611	167.0427	-16.8317	-86.1314	59.36592	26.1932	11.28255	-14.2541
65	-1.00127	-142.162	-41.4444	81.36544	8.191125	89.70648	123.0375	79.44361	71.68938	90.37859
66	79.77431	-48.4024	56.5247	118.5734	0.649405	39.48565	-21.5557	1.34839	79.30531	8.167277
67	-30.5595	26.72506	21.41718	27.58575	203.2099	108.593	-76.4798	-108.935	49.53174	2.456312
68	-107.456	-62.9912	96.44555	-23.1755	-10.3859	-18.3369	-10.1214	-22.1481	-108.884	42.30924
69	81.08529	9.206816	127.683	124.392	74.95708	-104.39	62.36478	-38.0946	9.647648	-7.56732
70	0.501912	22.35333	100.7672	-33.5171	-12.9747	14.99442	-74.9552	-22.6875	-32.6551	68.82603

	31	32	33	34	35	36	37	38	39	40
1	-27.593	26.30121	-40.2431	50.37821	13.62818	36.19276	-6.8837	-20.7582	-44.7982	53.46971
2	112.1829	78.63361	24.92122	-52.8093	-36.1533	16.17299	-22.7994	-30.1175	65.15543	-40.2844
3	-62.3895	12.89649	11.18797	30.14994	5.566292	93.75968	-46.4019	-25.0665	-13.7569	-27.2257
4	-100.434	67.42459	-20.1688	-6.38383	-63.4663	-8.11268	32.12533	-5.24411	36.2463	-76.8146
5	60.22013	-38.2879	6.31833	6.117059	58.34457	-10.7764	105.9962	-17.1467	33.91806	13.03027
6	68.53807	13.27372	-3.81534	5.42479	49.54579	-12.0545	15.58454	-11.4326	44.57806	40.58263
7	-16.5903	-17.9762	-21.0848	-73.099	-41.748	40.03626	-15.2939	-10.2717	22.31017	-20.3855
8	39.45003	-33.9129	-25.2911	37.24013	-13.0504	-8.38704	-33.0954	36.35967	-43.6386	-52.9485
9	-80.2539	37.38364	-34.8811	116.0562	-49.2099	26.62258	3.48501	-17.8342	92.49544	64.67377
10	42.77347	-35.6511	-90.6887	-25.4626	15.62892	-77.3801	-40.2535	1.597089	-3.54148	-4.93814
11	-32.5087	61.64298	1.383837	1.950672	-44.599	44.97121	59.15323	-10.7733	-38.6783	-29.8649
12	-11.3072	-15.7812	22.67738	51.79472	55.01014	-39.0828	61.37931	29.07744	2.429786	-16.6887
13	-23.6009	-30.7676	-9.66979	-52.3118	104.5129	-45.7275	6.573379	19.38152	23.72131	-3.22247
14	45.8233	24.70298	15.65627	34.41566	-13.305	49.75595	60.36433	-58.5256	-19.7925	32.68248
15	81.34159	165.61	-2.64343	-76.4833	-17.0563	-15.3999	-26.953	60.81901	-49.4183	-58.6868
16	-62.5489	30.54447	68.68791	10.2699	-32.4023	-46.5018	5.408414	-46.759	-6.27159	-4.33179
17	66.92376	-98.3327	-102.235	79.44322	1.817727	-57.6882	-36.6214	43.27681	-41.4684	8.556424
18	-49.3934	-61.0615	76.53672	-75.2221	-28.8464	42.79874	14.11591	85.17792	11.14843	-64.6613
19	-45.7698	-2.83194	23.61914	-113.376	65.13051	-38.2114	-23.886	-34.1192	-15.0958	5.027221
20	79.23843	57.09159	5.997397	65.40787	-59.9295	-97.0338	16.66491	17.9203	3.614809	29.58287
21	-5.42552	-5.52832	63.26677	-52.4067	-37.0558	86.02138	-4.00459	-68.1516	-33.2875	54.87242
22	-49.7226	69.31462	-76.4289	-91.8003	-56.6455	-51.8139	-5.73483	52.95862	-30.1504	26.22819
23	3.351388	30.96463	-5.83772	83.53373	113.1756	-13.9016	-7.5551	-9.82869	2.72659	10.34913
24	-33.1321	-26.0142	-21.0493	-6.30082	-4.5728	-5.75283	-11.1514	-3.34816	9.422114	-9.07757
25	-51.6909	6.845098	5.145891	59.67823	-22.9137	-21.8277	9.903207	-32.5304	69.19589	-10.1142
26	30.23361	11.12948	-37.884	-1.04432	78.5993	-43.2216	-21.4461	-13.6632	-43.5054	-28.4657
27	64.24789	-40.0156	-14.6194	7.592047	5.945346	13.07462	7.149053	2.204366	38.1859	51.16435
28	-3.03717	-52.7394	-57.9847	30.76567	-50.2594	-13.1282	-74.6484	-83.3944	-31.8387	-27.7253
29	12.13591	41.08097	53.92992	-56.018	8.846086	-70.6432	-19.8723	-41.0761	-57.6066	63.33383
30	92.85534	-66.4691	23.85727	-0.081	-8.00552	-0.57085	13.02588	-28.1364	-21.8504	-32.4059
31	11.44615	-0.81541	-39.9471	5.554965	15.3124	51.35239	-32.4044	-41.324	42.21385	-104.046
32	-4.18284	17.56481	-63.0673	-11.527	-7.54436	86.15355	59.5171	-24.706	-36.3189	20.48275
33	17.7061	-54.3198	51.53104	17.93791	16.84983	9.246197	26.70956	-3.93259	-25.8692	3.877239
34	55.79895	-47.2428	4.326675	-26.7931	12.3219	-52.0911	86.65051	35.9758	-44.6444	32.05906
35	31.6695	42.61518	-78.7711	-101.87	-24.4647	38.44326	-66.6536	54.30341	50.25	-3.84005
36	25.03288	5.228929	12.98585	7.337941	15.19522	70.80908	1.884463	48.10888	38.52667	-56.4377
37	-96.3282	39.69311	-23.2069	19.3632	-14.2872	22.21618	24.27042	19.44655	60.62881	1.032475
38	1.195178	-97.0403	24.27247	-39.7523	-96.4706	-48.005	24.4395	47.22102	-5.89729	57.7995
39	-7.08282	-15.4646	-134.599	33.82519	11.97974	80.91691	9.336316	0.264697	-108.732	28.38578
40	60.00008	14.61662	95.02514	-55.8813	-9.10892	-70.4543	-16.561	-11.6662	55.69354	-27.6152
41	-35.5629	67.22481	27.1658	18.49167	-31.8031	-31.8596	-97.3419	-22.845	19.40084	30.95638
42	-65.7691	-6.11557	-62.5274	43.59707	-28.2459	4.763257	-37.8854	-33.8196	-8.35288	52.70744
43	-53.1377	-38.696	-8.18089	-31.8838	81.70257	37.37219	-43.9833	56.09536	12.08413	27.3242
44	17.67272	31.00266	-47.3312	-113.286	7.41459	41.79498	68.46556	11.78186	46.2167	63.03184
45	-39.96	-24.1306	141.7148	47.21764	-39.4659	5.426806	16.25996	-23.3315	-27.9151	23.77697
46	54.05219	-79.3159	1.175209	-38.1044	41.08643	18.14783	-5.61371	46.92814	-16.6056	5.061166
47	61.59332	37.55171	9.873827	49.03722	-2.11335	-66.4235	-6.00764	-14.5214	32.1758	26.84139
48	6.057835	-79.0083	2.641423	-49.1903	-54.8219	-22.8712	40.32396	-111.041	42.34291	-53.8715
49	-26.1294	90.95097	112.5211	50.41705	-31.5769	-61.0363	59.95229	-45.7907	-69.5689	-42.7852
50	13.27116	-18.2019	88.15559	31.81571	119.1967	51.99843	-10.4656	11.09193	27.77654	-24.1074
51	44.39923	-2.5316	18.71548	97.49226	-45.6111	57.50934	-116.932	56.7218	35.194	-36.6678
52	-67.7843	-70.0238	151.6783	5.478471	86.41536	55.82193	0.489009	63.40785	3.446289	15.82766
53	-36.8964	-9.46203	-0.84447	73.58916	-32.8029	-81.4759	37.88034	63.19748	41.30535	36.50787
54	53.48664	-135.426	-63.0465	-30.8414	0.484382	32.01725	27.55547	-117.51	14.23736	-27.0383

	31	32	33	34	35	36	37	38	39	40
55	-148.622	18.75777	-27.0319	-96.7959	0.373772	-54.1404	37.44886	23.65788	-28.9034	32.12159
56	48.6353	0.795368	17.96009	59.76937	26.84766	-12.5612	-2.83547	23.33297	-4.97308	-21.4679
57	106.3499	29.66817	81.93055	-44.8785	-12.9886	28.44124	-13.4434	0.716224	-15.2092	36.42528
58	-69.0847	-6.58707	40.90517	6.371449	-17.4925	6.172719	-26.9563	-62.904	-72.1655	-10.3673
59	21.17423	112.4667	8.841067	0.998054	83.52577	28.42273	-78.167	-48.9318	25.03188	36.34325
60	45.74174	14.42198	20.24864	-31.4673	0.187781	71.20411	4.465199	27.85259	-26.1457	64.31644
61	2.324919	14.3207	43.52586	12.09971	12.0486	28.2252	-62.1702	16.5043	-52.4796	26.97085
62	-44.1288	-79.1852	56.33324	0.813965	-39.8418	-37.0703	-98.9206	79.27179	-30.1571	-11.2684
63	-41.0356	34.67717	-48.329	31.90358	15.22031	15.53222	-10.2711	75.26079	-60.5375	-38.457
64	-103.837	-57.7668	-29.6193	-31.7698	29.16008	9.807553	-4.69918	-38.8846	76.57324	36.76633
65	14.5442	-78.1752	-33.6309	15.04807	-20.4279	-68.8574	14.35542	-11.175	-3.21565	-62.332
66	36.44407	47.01646	-21.534	-9.54999	48.59481	-66.1621	-56.9647	-0.1373	63.32374	52.51302
67	123.0337	29.18039	6.899692	31.07079	-115.391	98.60435	70.83175	57.16265	14.45246	31.85499
68	26.70102	73.24744	-13.2039	-25.0171	78.51967	-12.1936	-13.1765	-96.2994	-46.1723	-41.8689
69	-6.88221	-55.8199	30.30351	-54.9611	-64.0502	-8.66059	-44.7815	4.279843	4.082571	-33.2896
70	-40.7879	80.79029	-52.5289	28.69889	64.23548	-6.43454	137.4958	60.88908	7.699731	-85.3459

	41	42	43	44	45	46
1	-10.4239	49.86426	57.04691	-31.1996	-53.7265	-26.7954
2	29.75103	-37.5342	15.38839	-7.45708	11.59359	-29.9039
3	-94.2741	-56.3173	-61.1008	-69.1563	-19.6882	26.43837
4	82.91849	55.43702	3.045723	-22.2325	28.22659	-24.0078
5	-18.4891	49.90461	7.516401	-43.3855	-23.5311	16.86118
6	-1.91158	-38.0437	42.0059	2.245646	-57.204	-12.8878
7	-28.8369	-64.8999	-10.2751	-1.8649	7.389404	-5.20276
8	-1.01977	3.629352	-21.7768	42.71381	-49.282	-19.1802
9	-22.1921	51.7882	5.945076	47.77301	64.40328	23.99169
10	20.93086	-17.7444	-22.1531	23.25206	-0.49517	26.8725
11	11.15979	-3.89594	49.56466	12.45209	-71.7835	-24.3884
12	41.80462	-10.4313	-12.0429	6.617438	43.69218	24.51387
13	-3.49154	-30.3783	41.46067	-39.9949	0.259308	1.893635
14	-16.705	-45.2816	38.25941	34.83018	14.91831	-25.9291
15	3.061004	34.28767	-26.8756	60.94583	17.59736	7.455215
16	-3.61598	41.50188	-21.1338	86.69684	12.55293	25.22638
17	-39.1513	11.82484	-24.7267	10.78147	-64.0756	20.86709
18	-40.1099	59.15295	54.44608	39.70587	59.87658	-18.5665
19	39.33823	4.985222	49.68908	-66.4157	17.85746	18.53987
20	-7.83772	-0.10032	-58.6942	-57.3402	88.18923	11.30887
21	8.272563	-36.5638	-44.5729	-20.9545	44.16768	18.86039
22	-71.9858	16.32788	87.14776	-31.0574	-30.6903	27.45133
23	-37.5585	-45.4	33.64424	18.21313	30.06863	-46.1482
24	-0.70586	-5.47284	-3.77509	9.743969	2.928133	-12.7351
25	11.31988	-31.3369	-23.3058	-7.05733	-34.469	18.10748
26	12.85883	34.38641	6.608118	4.855143	58.06356	16.46351
27	-65.8547	-11.0578	-0.87418	-6.31876	5.208178	10.01251
28	-17.468	24.80223	72.70128	-30.0711	-2.87934	10.67146
29	-21.238	15.73925	-27.5482	-89.3776	43.05094	-35.7797
30	-5.9318	36.56952	-109.27	14.4066	-28.9706	-28.1034
31	64.18109	26.09314	-29.5021	-43.4502	-13.3129	-35.4261
32	30.33509	-28.2729	0.926494	41.84903	-6.75457	50.32769
33	41.18389	54.77593	-0.93082	26.56568	19.14448	53.5048
34	-15.3301	15.11267	-36.8612	-8.21216	-26.0087	-29.2739
35	40.53403	-14.8201	-34.8792	-24.9922	-26.1388	52.76558
36	-33.8055	24.54753	27.23779	10.83246	13.80538	27.47657
37	3.716348	28.67619	-46.9275	-23.4948	-24.962	24.30542
38	26.34711	41.53799	23.26854	30.8588	-11.4057	-58.7994
39	55.36693	17.12332	22.11017	-14.7854	51.76	24.12836
40	-33.8413	-7.53234	13.0851	12.6702	-25.2119	-22.8836
41	9.819837	5.308736	-5.33871	62.21594	20.42486	27.16466
42	18.89564	-17.7068	-11.1807	-22.8571	24.71273	-91.9593
43	-56.3417	53.30859	-21.2645	22.60134	62.8209	-42.2011
44	-28.968	16.9436	-75.3057	38.45113	-34.0507	41.55841
45	-21.8295	16.6675	-5.69652	1.88764	-5.01813	-10.9217
46	64.3658	-18.0765	21.24391	27.86025	17.94155	34.00698
47	121.6007	-24.0951	-28.3666	-21.7279	-23.2896	-47.8844
48	-35.0686	-3.54764	-17.8653	-16.2842	-26.2886	-38.8884
49	15.76763	-64.9337	55.42515	15.17469	-31.0183	79.12297
50	6.373278	-49.6691	-44.822	37.07413	16.01078	19.15862
51	26.78905	12.96885	45.31169	-77.3287	-23.1776	38.73689
52	28.90476	40.7979	20.72949	-53.4521	-72.0079	28.88653
53	-13.5466	0.442944	-48.2188	-41.4158	10.61992	74.42201
54	43.81422	8.765946	27.23681	39.81024	19.77933	24.29135

	41	42	43	44	45	46
55	23.05212	-37.551	-28.5243	-9.46702	-55.8826	-3.31033
56	-11.8561	46.51602	10.38893	-43.6186	8.950044	-30.9574
57	52.55131	68.30861	13.95811	-33.3762	6.08608	-2.55329
58	16.89688	53.89413	-31.1063	29.01262	-9.83101	-8.49292
59	-69.5414	49.66116	11.47537	54.48243	-57.0433	-57.2726
60	26.37746	-56.332	22.89101	-12.6793	63.33224	-18.4055
61	7.442063	-109.728	-23.7737	23.0318	-11.4155	-8.52699
62	55.89654	-59.4547	-11.5191	32.2933	-13.9372	-25.6394
63	-6.51827	12.81742	-74.8843	12.11249	-22.5178	-40.677
64	56.9889	-14.1698	-16.1881	7.744763	0.855971	-18.6241
65	-78.0362	-50.384	43.61897	22.35945	16.0932	21.453
66	31.19062	8.887882	58.20883	47.79926	-1.86948	24.33102
67	-10.9233	-4.28202	10.24767	-15.5563	-18.8046	-12.0301
68	-27.6089	47.00775	-54.7129	-37.0588	-1.01989	55.31901
69	-73.8015	-30.6556	-10.9322	-35.5165	74.60915	5.166553
70	2.717575	-48.0955	17.78833	-14.2625	43.73626	-66.0222

Bibliografia

[1] Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning from data

[2] Casali, M., Gagliardi, C., Grasselli, L.: Geometria. Progetto Leonardo. (2001), https://books.google.it/books?id=rOwyAAAACAAJ

[3] Fagertun, J.: Face Recognition. Ph.D. thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark (2005)

[4] Fischler, M.A., Elschlager, R.A.: The representation and matching of pictorial structures. IEEE Transactions on computers 22(1), 67–92 (1973)

[5] Goldstein, A.J., Harmon, L.D., Lesk, A.B.: Identification of human faces. Proceedings of the IEEE 59(5), 748–760 (1971)

[6] Huang, K.: Principal component analysis in the eigenface technique for facial recognition (2012)

[7] Jain, A.K., Li, S.Z.: Handbook of face recognition. Springer (2011)

[8] Pascucci, A.: Lezioni di probabilità e statistica (2016), http://www.dm.unibo.it/ pascucci/

[9] Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Josa a 4(3), 519–524 (1987)

[10] Strohminger, N., Gray, K., Chituc, V., Heffner, J., Schein, C., Heagins, T.B.: The mr2: A multi-racial, mega-resolution database of facial stimuli. Behavior research methods pp. 1–8 (2015)

[11] Thorat, S., Nayak, S., Dandale, J.P.: Facial recognition technology: An analysis with scope in india. arXiv preprint arXiv:1005.4263 (2010)

[12] Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. pp. 586–591. IEEE (1991)