



**UNIVERSITE D'ANTANANARIVO**  
-----  
**ECOLE SUPERIEURE POLYTECHNIQUE**  
-----  
**DEPARTEMENT TELECOMMUNICATION**



**MEMOIRE DE FIN D'ETUDES**  
  
en vue de l'obtention du  
  
**DIPLÔME de**  
  
**MASTER A VISEE PROFESSIONNELLE**

*Domaine* : Sciences de l'Ingénieur

*Mention* : TELECOMMUNICATION

*Parcours* : Systèmes de Traitement de l'Information (STI)

*par* : **RANDRIAMAHANDRY Vonjinirina Eric**

***RECONNAISSANCE FACIALE PAR  
METHODE ACP HYBRIDE***

Soutenu le 26 mai 2016 devant la Commission d'Examen composée de :

Président : M. ANDRIAMIASY Zidora Valinjaosera

Examineurs :  
M. RATSIMBAZAFY Andriamanga  
M. RANDRIAMITANTSOA Andry Auguste  
Mme RAMAFIARISON Malalatiana Hajaso

Directeur de mémoire : Mme ANDRIANTSILAVO Haja Samiarivonjy



## **REMERCIEMENTS**

Je rends grâce à Dieu pour sa bénédiction de m'avoir toujours donné la force et la santé durant mes études, mais aussi de mener à bien la réalisation de ce travail de mémoire de fin d'études et sans qui tout cela n'aurait été possible.

Je voudrais remercier Monsieur ANDRIANARY Philippe Antoine, Professeur Titulaire, Directeur de l'École Supérieure Polytechnique d'Antananarivo, qui a bien voulu m'accueillir dans cet établissement.

J'associe à ces remerciements, Monsieur RAKOTOMALALA Mamy Alain, Maître de Conférences, Chef de Département Télécommunication.

Je remercie sincèrement Monsieur ANDRIAMIASY Zidora Valinjaosera, Maître de Conférences, Enseignant au sein du Département Télécommunication, qui nous a fait l'honneur de présider le Jury de ce mémoire.

Je suis très reconnaissant envers Madame ANDRIANTSILAVO Haja Samiarivonjy, Enseignante au sein du Département Télécommunication, qui m'a encadrée et sans laquelle mon travail et l'aboutissement de ce mémoire n'auraient pas vu le jour.

J'exprime également ma gratitude aux membres de jury qui ont accepté d'examiner ce mémoire malgré leurs innombrables occupations :

- Monsieur RATSIMBAZAFY Andriamanga, Maître de Conférences, Enseignant chercheur au sein du Département Télécommunication.
- Monsieur RANDRIAMITANTSOA Andry Auguste, Maître de Conférences, Enseignant chercheur au sein du Département Télécommunication.
- Madame RAMAFIARISON Malalatiana Hajaso, Maître de Conférences, Enseignante chercheuse au sein du Département Télécommunication.

Ce travail de mémoire n'aurait pu être mené de façon efficace et rigoureuse en parallèle à ma formation sans l'aide des différents enseignants et personnel administratif de l'École, à qui j'adresse toute ma gratitude.

Enfin, je n'oublie pas mes plus sincères remerciements à toute ma famille pour leur contribution, leur soutien et leur patience, ainsi qu'à tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

## TABLE DES MATIÈRES

REMERCIEMENTS.....	i
TABLE DES MATIÈRES .....	ii
NOTATIONS ET ABRÉVIATIONS.....	v
INTRODUCTION GENERALE.....	1
CHAPITRE 1 SYSTEMES BIOMETRIQUES.....	2
1.1 Généralités .....	2
1.1.1 Introduction.....	2
1.1.2 Définition.....	2
1.1.3 Historique .....	3
1.2 Architectures d'un système de reconnaissance biométrique .....	3
1.3 Performances des systèmes biométriques .....	4
1.3.1 Système d'évaluation de la performance .....	5
1.3.2 Seuil de décision .....	6
1.4 Applications biométriques .....	8
1.4.1 Empreinte Digitale .....	8
1.4.2 Forme de la main .....	9
1.4.3 Réseau veineux.....	9
1.4.4 Examen de l'œil.....	10
1.4.5 Signature .....	12
1.4.6 Reconnaissance vocale.....	13
1.4.7 Reconnaissance faciale .....	14
1.5 Conclusion.....	15
CHAPITRE 2 RECONNAISSANCE FACIALE .....	16
2.1 Généralités .....	16
2.1.1 Introduction.....	16
2.1.2 Définition.....	16
2.1.3 Processus de reconnaissance faciale .....	16
2.2 Approches de la détection de visage .....	18
2.2.1 Méthodes basées sur l'apparence .....	18
2.2.2 Méthodes basées sur les connaissances acquises .....	19

2.2.3 Méthodes basées sur l'appariement de gabarits « Template-matching » .....	19
2.2.4 Méthodes basées sur les caractéristiques du visage .....	20
<b>2.3 Algorithme de reconnaissance faciale.....</b>	<b>21</b>
2.3.1 Méthodes globales .....	21
2.3.2 Méthodes locales .....	23
2.3.3 Méthodes hybrides.....	24
<b>2.4 Conclusion.....</b>	<b>25</b>
<b>CHAPITRE 3 ANALYSE EN COMPOSANTES PRINCIPALES .....</b>	<b>26</b>
<b>3.1 Présentation .....</b>	<b>26</b>
3.1.1 Introduction et définition .....	26
3.1.2 Brève histoire de l'ACP.....	26
<b>3.2 Transformation unitaire : Images et ACP .....</b>	<b>27</b>
3.2.1 Etape 1 : Numériser et vectoriser des images .....	28
3.2.2 Etape 2 : Calculer la matrice de produits croisés.....	29
3.2.3 Etape 3 : Diagonaliser la matrice de produits croisés.....	29
3.2.4 Etape 4 : Calcul des projections des images .....	30
3.2.5 Etape 5 : Reconstruction des images initiales .....	31
3.2.6 Image supplémentaire .....	33
<b>3.3 ACP en reconnaissance faciale .....</b>	<b>34</b>
3.3.1 Principes et méthodes eigenfaces.....	34
3.3.2 Projection des images de visage .....	38
3.3.3 Phase d'apprentissage.....	39
3.3.4 Phase de test .....	40
<b>3.4 Conclusion.....</b>	<b>42</b>
<b>CHAPITRE 4 SIMULATION RECONNAISSANCE FACIALE HYBRIDE.....</b>	<b>43</b>
<b>4.1 Présentation du logiciel MATLAB.....</b>	<b>43</b>
4.1.1 Introduction.....	43
4.1.2 Environnement Matlab .....	44
<b>4.2 Commande de base dans Matlab .....</b>	<b>45</b>
4.2.1 Aide dans MATLAB .....	45
4.2.2 Fonctions sur les vecteurs et matrices .....	45
<b>4.3 Détections du visage et les parties du visage .....</b>	<b>46</b>

4.3.1 Principe de l'algorithme Viola Jones .....	46
4.3.2 Détection de visage avec la fonction : <i>CascadeObjectDetector</i> .....	49
<b>4.4 ACP sous MATLAB.....</b>	<b>50</b>
4.4.1 Fonction <i>Eig</i> .....	50
4.4.2 Distance euclidienne .....	51
<b>4.5 Reconnaissance de visage, des yeux et du nez.....</b>	<b>51</b>
4.5.1 Organigramme du programme .....	51
4.5.2 Base de visage standard: Collection d'images du visage <i>Faces94</i> .....	54
4.5.3 Base de visage personnalisée .....	55
4.5.4 Interface de notre système de reconnaissance.....	56
4.5.5 Résultat des simulations et statistique d'erreur.....	57
<b>4.6 Conclusion.....</b>	<b>59</b>
<b>CONCLUSION GENERALE .....</b>	<b>60</b>
<b>ANNEXE 1 OPERATIONS SUR LES MATRICES ET VECTEURS.....</b>	<b>61</b>
<b>ANNEXE 2 BASE DE DONNEES DE VISAGE .....</b>	<b>66</b>
<b>ANNEXE 3 CODE DE L'ACP SOUS MATLAB.....</b>	<b>71</b>
<b>BIBLIOGRAPHIE .....</b>	<b>73</b>
<b>FICHE DE RENSEIGNEMENTS .....</b>	<b>75</b>

## NOTATIONS ET ABRÉVIATIONS

### 1. Minuscules latines

cm	Centimètres
d	Distance
$g_i$	Poids des images sur l'i-ème vecteur propre
$u_i, e_i, v_i$	i-ème vecteurs propres
$x_i$	Vecteur représentant les images obtenues après $\text{vec}(Y_i)$

### 2. Majuscules latines

ABCD	Sommets d'un rectangle
$C_i$	Capture image
C	Matrice de covariance
D	Dimension de l'espace vectoriel
G	Matrice projections des images
$G^T$	Transposé de la matrice G
$H_0$	Hypothèse que $C_i = \text{imposteur}$
$H_1$	Hypothèse que $C_i = \text{utilisateur légitime}$
$I_i(m, n)$	Représentation matricielle d'une image de dimension $m \times n$
$L_p$	Distance de Minkowski d'ordre p
$\text{Mah}_{L2}$	Distance Mahalanobis $L2$
N	Nombre d'images d'apprentissage
P	Probabilité
U, A	Matrice rassemblant les vecteurs propres
$U^T, A^T$	Transposé de la matrice U et A
X	Représentation matricielle d'un ensemble d'images
$X^T$	Transposé de la matrice X
$Y_i$	Matrice représentant des images $3 \times 3$
Z	Représentation intégrale d'un sommet d'un rectangle

### 3. Minuscules Grecs

$\lambda_i, \mu_i$	Valeurs propres
$\mu m$	Micromètres

$\sigma_i$	Ecart-type
------------	------------

#### 4. Majuscules Grecs

$\Lambda$	Valeurs propres d'une matrice
$\Phi$	Différence entre toutes images et image moyenne
$\Psi$	Image moyenne de toutes images collectées
$\Gamma_i (m \times n, 1)$	Représentation vectorielle d'une image

#### 5. Abréviations

2-D	2-Dimension
3-D	3-Dimension
AAM	Active Appearance Model
ACI	Analyse en Composantes indépendantes
ACL	Analyse de Caractéristiques Locales
ACP	Analyse en Composantes Principales
ADL	Analyse Discriminante Linéaire
ASR	Automatic Speech Recognition
BIC	Bayesian Information Criterion
DET	Detection Error Trade-off
DSP Blockset	Digital Signal Processing Blockset
EBGM	Elastic Bunch Graph Matching
EER	Equal Error Rate
EO	EigenObject
FAR	False Acceptance Rate
FRR	False Rejection Rate
HMM	Hidden Markov Models
IBM	International Business Machines
ICA	Independent Component Analysis
ID	Identité
JPG	Joint Photographic Experts Group
K-LDA	Kernet LDA
K-PCA	Kernel PCA
LBP	Locally Binary Patterns



LDA	Linear Discriminate Analysis
LED	Light Emitting Diode
LFA	Local Feature Analysis
LG-ACP	Log Gabor ACP
MATLAB	MATrix LABoratory
Mhz	Mégahertz
NN	Neuronal Network
OpenCV	Open Source Computer Vision
PC	Personnal Computer
PCA	Principal Components Analysis
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines (machine à vecteurs de support)
TFA	Taux de Fausses Acceptations
XMLFILE	Extensible Markup Language-FILE

## INTRODUCTION GENERALE

La large disponibilité des ordinateurs de bureau et portables puissants et à bas prix, et des systèmes informatiques embarqués a créé un intérêt énorme dans le cadre du traitement automatique d'images numériques et vidéos dans plusieurs applications, y compris l'authentification biométrique, la surveillance, l'interaction homme-ordinateur et les gestions multimédias. Les recherches et développements sur les systèmes de reconnaissance automatique du visage suivent naturellement.

Reconnaître des personnes est une tâche que les êtres humains exécutent régulièrement et facilement dans leurs vies quotidiennes. Et la finalité de la reconnaissance faciale est de mettre au point des systèmes informatiques capables d'égaliser cette faculté humaine.

En ce moment, il existe plusieurs variétés d'algorithmes de reconnaissance faciale alors que les premiers modèles de reconnaissances automatiques ou semi-automatiques des visages représentaient les visages comme des vecteurs de distances, angles, ou rapports entre différents traits constituant les visages comme les yeux, le nez et la bouche. En reprenant ce concept et en l'utilisant avec une méthode d'analyse de donnée matricielle, le résultat devient de plus en plus acceptable. L'ACP est l'un des meilleurs systèmes d'analyse de données. L'idée est de faire à la fois la reconnaissance par ACP sur le visage et sur les traits constituant du visage. C'est ce qui nous a amenés à intituler ce mémoire : « Reconnaissance faciale par méthode ACP-hybride ».

Pour cela, nous abordons dans le premier chapitre la généralité sur les systèmes biométriques suivis de la reconnaissance faciale dans le second. Dans le troisième chapitre se présente l'Analyse en Composant Principale, le modèle mathématique de notre algorithme. Pour le dernier, la conception du logiciel, la réalisation sur Matlab, ainsi que les résultats, c'est ce qui renferme cette partie simulation.

# **CHAPITRE 1**

## **SYSTEMES BIOMETRIQUES**

### **1.1 Généralités**

#### ***1.1.1 Introduction***

De nos jours, l'accroissement des terminaux d'accès et l'évolution des moyens de communication dans les divers systèmes tels que les déplacements physiques, les transactions financières, l'accès aux services et bien d'autres encore met en cause le besoin de s'assurer de l'identité des individus. Les moyens classiques de vérification d'identité pour les contrôles d'accès, comme les passeports ou cartes d'identité, les mots de passe ou les codes secrets peuvent être facilement falsifiés. Il serait aussi possible que les utilisateurs perdent leur carte d'identité, oublient leur mot de passe, etc.

La solution évidente pour compenser ces inconvénients est d'utiliser la biométrie tout en basant la vérification sur les aspects propres de chaque individu. Cette technologie est maintenant exploitée pour un très grand nombre d'applications [1].

Elle peut être appliquée dans le système de contrôle d'accès physique à des sites sensibles comme les entreprises, les aéroports, ou les banques ou bien dans le système de contrôle d'accès logique à un ordinateur, ou site web. A l'heure actuelle, dans certains endroits, elle est déjà appliquée dans le domaine de la vidéo surveillance qui utilise les technologies de l'identification à distance et en mouvement.

#### ***1.1.2 Définition***

La biométrie peut être définie comme étant : « la reconnaissance automatique d'une personne en utilisant des traits distinctifs ». On la définit aussi comme « toutes caractéristiques physiques ou traits personnels automatiquement mesurables, robustes et distinctifs qui peuvent être utilisés pour identifier un individu ou pour vérifier l'identité prétendue d'un individu » [2].

La biométrie propose beaucoup plus d'avantage que les méthodes d'authentification personnelle existantes. Les caractéristiques biométriques sont propres et uniques à chaque individu et la possibilité que deux individus aient exactement les mêmes caractéristiques biométriques est peu probable. En effet, elle apporte encore plus de sûreté et comble les grands handicaps de sécurité par mots de passe.

### ***1.1.3 Historique***

Le terme le plus exact pour décrire le champ de la biométrie serait sans doute celui d'anthropométrie (du grec anthropo, « homme », et mitron, « mesure »). Alphonse Bertillon (1853-1914), le fondateur de la police scientifique en 1880, avait recours à des données biométriques lorsqu'il traquait les récidivistes par le bertillonnage, en mesurant diverses parties du corps ou du visage qui ne change pas à l'âge adulte. Quelques années plus tard, en 1892, le Britannique Francis Galton (1822-1911) allait révolutionner les méthodes policières en démontrant par la statistique le caractère unique des empreintes digitales. Aujourd'hui, le progrès des sciences et des techniques permet d'autres modes d'identification des personnes. L'informatique évince les fichiers manuels. Elle permet non seulement des traitements accélérés, mais aussi des contrôles à distance. L'identification et l'authentification des personnes sont ainsi facilitées.

L'introduction de données biométriques a pour but de renforcer la sécurité des Etats à l'heure de la mondialisation. La menace terroriste affecte une particulière acuité : les attentats du 11 septembre 2001 aux États-Unis et du 11 mars 2004 en Espagne en sont l'illustration. La criminalité organisée et l'immigration illégale justifient également la prise de mesures particulières de protection.

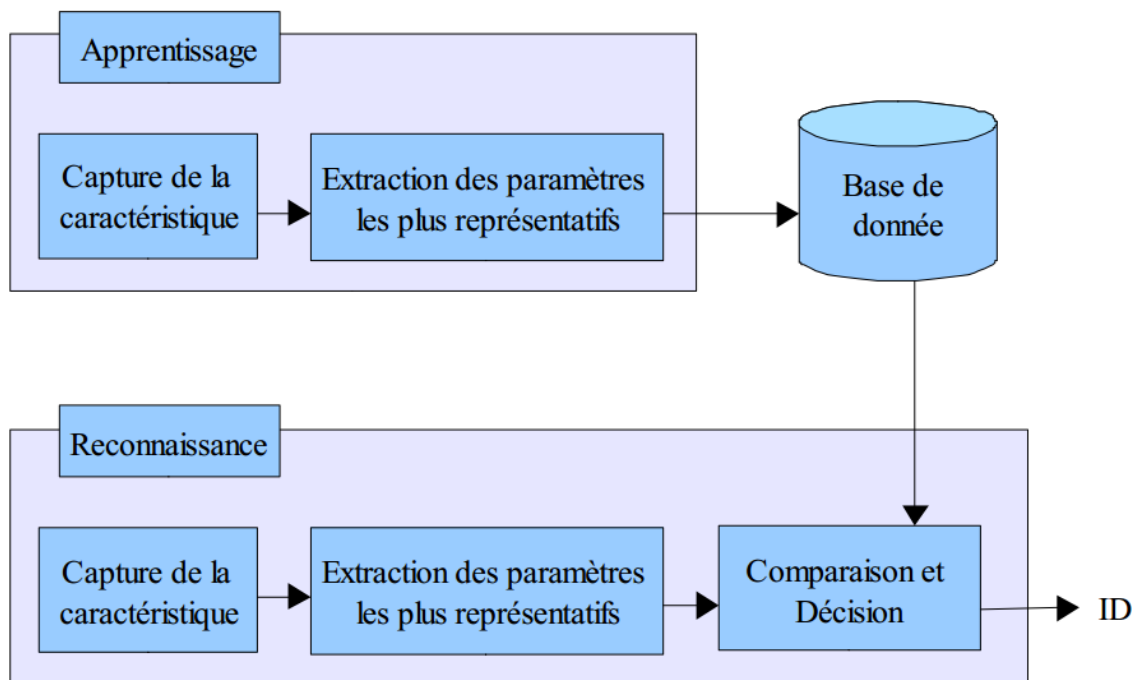
La biométrie est de plus en plus utilisée dans les aéroports, les établissements pénitentiaires, pour l'accès à des locaux sécurisés, pour la garantie du vote électronique, la sécurité des paiements bancaires ou des transactions via Internet. Le marché associé à son développement est estimé à 900 millions de dollars pour 2006. Les enjeux économiques sont donc très importants et pèsent sur le choix des normes et des matériels en vue d'une interopérabilité. L'amélioration de la sécurité qui résulte du recours à la biométrie ne doit pas pour autant porter atteinte aux libertés individuelles [24].

## **1.2 Architectures d'un système de reconnaissance biométrique**

En général, un système biométrique se compose de deux phases : une phase d'apprentissage et une phase de reconnaissance.

Tout d'abord, on enregistre la caractéristique biométrique (l'empreinte digitale, l'iris, ou la signature, etc.) à l'aide d'un capteur. En effet, on n'extrait que les paramètres dits « pertinents » de la donnée, ce qui permet de réduire significativement la taille des données à sauvegarder vu que certaines informations sont inutiles à la reconnaissance. De plus, à partir de ces paramètres, il est impossible de revenir au signal original.

On peut voir dans cette figure 1.01 l'architecture générale d'un système de reconnaissance biométrique.



**Figure 1.01 :** *Architecture d'un système de reconnaissance biométrique*

La phase d'apprentissage permet de constituer le modèle d'une personne donnée à partir d'un ou plusieurs enregistrements de la caractéristique biométrique considérée. Ces modèles sont ensuite enregistrés dans une base de données ou sur une carte de type carte à puce.

Au cours de la phase de reconnaissance, la caractéristique biométrique est capturée et les paramètres pertinents sont extraits comme dans la phase d'apprentissage. La suite de la reconnaissance dépend du mode opératoire du système. Si on est en mode identification, le système va comparer le signal capturé avec tous les modèles contenus dans la base de données. Puis, il va tirer le modèle le plus proche du signal pour répondre à la question du type : « Qui suis-je ? ». C'est une tâche très difficile, car la base de données peut contenir des milliers d'individus. On perd beaucoup de temps pour calculer toutes les comparaisons possibles. Par contre, en mode vérification, le système va comparer ce signal avec un seul des modèles dans la base de données (par exemple une carte à puce) avec pour but de répondre à la question : « Suis-je bien la personne que je prétends être ? » [3].

### 1.3 Performances des systèmes biométriques

La performance d'un système biométrique est le critère le plus important lors de sa conception. Un système non performant serait une simple futilité.

### 1.3.1 Système d'évaluation de la performance

La performance d'un système biométrique est évaluée principalement selon trois critères : sa précision, son efficacité (c'est-à-dire sa vitesse d'exécution) et le volume de données qui doit être stocké pour chaque locuteur.

On a vu précédemment qu'il existe deux modes opératoires différents dans le système biométrique : le mode identification et le mode vérification.

#### 1.3.1.1 Evaluation du mode identification

Dans ce premier mode opératoire, le taux d'identification est la mesure la plus communément utilisée, mais il reste toujours insuffisant. En effet, en cas d'erreur, il peut être indispensable de savoir si les N premiers résultats incluant le bon choix. On trace alors le score cumulé (cumulatif match score) qui exprime la probabilité que le bon choix se trouve parmi les N premiers [3].

Dans le cas où il existe plusieurs modèles pour chaque individu dans la base de données, les mesures classiques des systèmes de recherche dans une base de données (database retrieval system) peuvent être utilisées [2].

La précision (precision) est le rapport entre le nombre de modèles correctement retrouvés par le système dans la base de données et le nombre total de modèles retrouvés.

$$\textbf{Precision} = \frac{N_{\text{correctement retrouvé}}}{N_{\text{retrouvé}}} \quad (1.01)$$

Le rappel (recall) est le rapport entre le nombre de modèles correctement retrouvés dans la base de données et le nombre total de modèles qui auraient dû être retrouvés.

$$\textbf{recall} = \frac{N_{\text{correctement retrouvé}}}{N_{\text{auraient dû être trouvé}}} \quad (1.02)$$

#### 1.3.1.2 Evaluation du mode vérification

Lorsqu'un système fonctionne en mode vérification, celui-ci peut faire deux types d'erreurs. Il peut rejeter un utilisateur légitime et dans ce premier cas on parle de faux rejet (false rejection). Il peut aussi accepter un imposteur et on parle dans ce second cas de fausse acceptation (false acceptance). La performance du système se mesure alors par son taux de faux rejet FRR (False Rejection Rate) et à son taux de fausse acceptation FAR (False Acceptance Rate) [4].

### 1.3.2 Seuil de décision

La vérification est un problème de décision similaire à la détection d'un signal dans le bruit en théorie de l'information. Il peut être formulé de la manière suivante. Soit  $H_0$  l'hypothèse : « la capture  $C_i$  provient d'un imposteur » et  $H_1$  l'hypothèse : « la capture  $C_i$  provient de l'utilisateur légitime ». Il faut donc choisir l'hypothèse la plus probable. On considère que la capture  $C_i$  provient d'un utilisateur légitime si :

$$P(H_1/C_i) > P(H_0/C_i) \quad (1.03)$$

En appliquant le théorème de Bayes qu'on peut voir dans [5], on obtient :

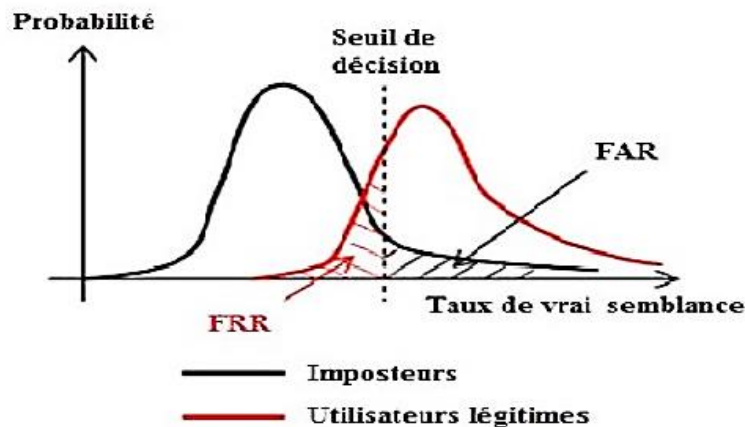
$$\frac{P(C_i/H_1)P(H_1)}{P(C_i)} > \frac{P(C_i/H_0)P(H_0)}{P(C_i)} \quad (1.04)$$

Et donc

$$\frac{P(C_i/H_1)}{P(C_i/H_0)} > \frac{P(H_0)}{P(H_1)} \quad (1.05)$$

Le taux de vraisemblance  $\frac{P(C_i/H_1)}{P(C_i/H_0)}$  est comparé à un seuil  $\Theta$  appelé seuil de décision. Les valeurs  $P(H_0)$  et  $P(H_1)$  qui représentent respectivement la probabilité pour qu'un imposteur ou un utilisateur légitime essayent d'accéder au système sont des valeurs difficiles à estimer.

Nous avons représenté sur la figure 1.02, la distribution hypothétique des taux de vraisemblance qu'obtiendraient les utilisateurs légitimes et les imposteurs d'un système de vérification donné.

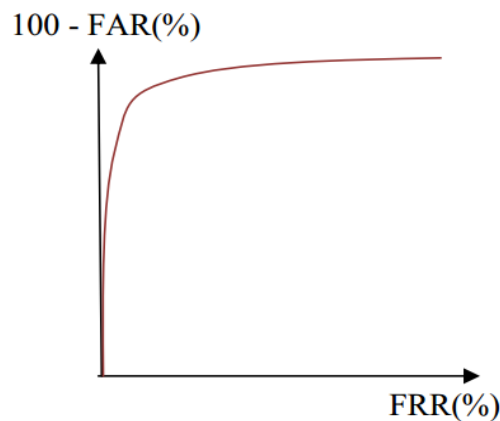


**Figure 1.02 :** Distributions des taux de vraisemblance des utilisateurs légitimes et des imposteurs d'un système de vérification biométrique.

Les FAR et FRR sont représentés en hachure. Le système idéal devrait avoir des FAR et FRR égaux à zéro. Comme ce n'est jamais le cas en pratique, il faut trouver un compromis entre FAR et FRR.

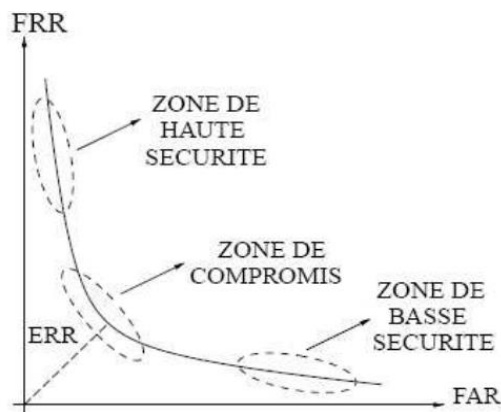
Plus le seuil de décision  $\Theta$  est bas, plus le système acceptera d'utilisateurs légitimes, mais plus acceptera aussi d'imposteurs. Inversement, plus le seuil de décision est élevé, plus le système rejettera d'imposteurs, mais plus il rejettera aussi d'utilisateurs légitimes [2] [6]. Il est donc impossible de faire diminuer les deux types d'erreurs en même temps en faisant varier le seuil de décision.

L'utilisation de la courbe ROC est aussi un outil d'évaluation de vérification biométrique très fréquent. Elle dessine la variation du taux de vérification ( $100 - \text{FAR}(\%)$ ) en fonction du taux FRR(%). La figure 1.03 montre un exemple de courbe ROC [6].



**Figure 1.03 : La courbe ROC**

Cependant une courbe ROC possède ses limites. Elle devient asymptotique à partir d'une certaine valeur du FRR lorsque celui-ci est très fort ou très faible. De manière générale, seulement 10% de cette courbe est utile et informative. C'est pour cela qu'on utilise les courbes DET qui représentent directement le taux FAR (%) en fonction du taux FRR (%) relativement aux courbes ROC. La figure 1.04 montre un exemple de courbe DET avec une zone de haute sécurité, une zone de basse sécurité et une de zone de compromis.



**Figure 1.04 : La courbe DET**



Plus la courbe DET est asymptotique aux grandes valeurs des deux axes et proche de l'origine, plus le système étudié a de bonnes performances. Concernant celle-ci le point le plus intéressant est l'EER qui correspond au point où les taux FRR et FAR sont égaux. Il correspond aussi à un choix de seuil qui est indépendant de l'utilisation du système. On utilise souvent ce point pour comparer la performance des différents systèmes biométriques même s'ils ont été évalués sur des bases de données différentes [3].

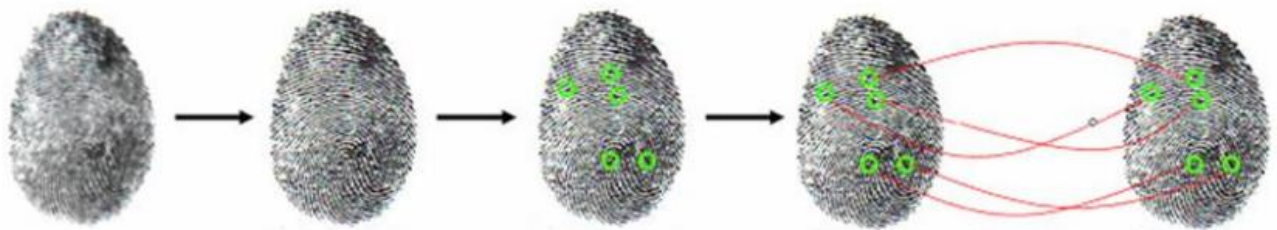
## 1.4 Applications biométriques

Aucun système biométrique n'est parfait, car chaque système a ses forces et ses faiblesses. Mais au moins, il y a certains d'entre eux qui sont plus performants par rapport aux autres selon l'application.

### 1.4.1 Empreinte Digitale

La reconnaissance des empreintes digitales est la méthode biométrique la plus répandue de nos jours. Les empreintes digitales sont composées de lignes localement parallèles présentant des points singuliers (minuties) et constituent un motif unique, universel et permanent.

Les capteurs utilisés dans ce système fonctionnent selon différents mécanismes de mesure (pression, champ électrique, température) qui permettent d'obtenir l'empreinte d'un doigt fixe positionné sur ce dernier (capteur matriciel) ou en mouvement (capteur à balayage).



**Figure 1.05 :** *Le processus de reconnaissance par empreinte digitale*

Après l'extraction des caractéristiques de l'empreinte de l'individu, le système crée le modèle approprié ; puis, les techniques d'appariement, qui peuvent être classées en deux catégories : les techniques basées sur la détection locale des minuties et les techniques basées sur la corrélation.

Vue la difficulté d'extraction des points de minutie sur une image de qualité médiocre, la méthode basée sur la corrélation est capable d'éviter ce problème en utilisant la structure globale de l'empreinte. Mais les résultats de la méthode par corrélation sont moins précis qu'avec les minuties, ce qui a donné l'idée de combiner les deux méthodes pour un système plus performant.

### ***1.4.2 Forme de la main***

La géométrie de la main est une technologie biométrique récente. Comme son nom l'indique, elle consiste à analyser et à mesurer la forme de la main, c'est-à-dire mesurer la longueur, la largeur et la hauteur de la main d'un individu et créer une image 3-D. Le capteur utilisé pour acquérir ses données est composé des LEDs infrarouges et d'un appareil photo numérique.

Cette technologie est généralement utilisée pour des systèmes comprenant l'enregistrement de présence et le contrôle d'accès parce qu'elle peut être facilement trompée par des jumeaux ou par des personnes ayant des formes de la main proches.



**Figure 1.06 :** *Dispositif de reconnaissance par géométrie de la main*

### ***1.4.3 Réseau veineux***

Relativement nouveau, cette méthode semble séduire. Elle se profile toutefois comme complément, par exemple à la géométrie de la main ou à l'empreinte digitale. On présente ce système biométrie dans [25].

#### ***1.4.3.1 Technologie***

Le sang chargé en oxygène arrive dans les mains par les artères, puis repart vers le cœur par les veines. Le sang appauvri en oxygène n'a pas le même filtre d'absorption que le sang artériel : il absorbe la lumière à des longueurs d'onde proches de l'infrarouge (autour de 760 $\mu$ m).

Quand la main est éclairée avec une lumière infrarouge, le réseau veineux apparaît en noir. Il est enregistré sous forme de « carte d'identité » dans une base de données, et pourra ensuite servir de comparaison lors de l'authentification.

Il existe deux types de « photographie » : la méthode « réflective » et la méthode par transmission. La méthode par transmission consiste à capturer la lumière qui passe à travers la main. En effet, quand il fait froid, les veines se contractent et le contraste n'est pas assez visible par transmission.

Par contre, dans la méthode réfective, la lumière reflétée n'est pas modifiée même par un temps froid. D'autre part, les deux parties du dispositif sont au même endroit, puisque la lumière est renvoyée. C'est donc plus facile à intégrer dans un petit appareil, par exemple un téléphone portable.

#### 1.4.3.2 Avantages et inconvénients

Il n'existe aucun moyen de frauder, car on ne peut pas « photographier » les veines de la main. Le réseau vasculaire est propre à chaque individu : même les vrais jumeaux en ont un différent. Selon Fujitsu, qui a comparé 140.000 paumes de la main, le TFA-taux de fausses acceptations (personnes authentifiées comme quelqu'un d'autre) est de moins de 0,00008 %. Il n'y a aucun contact lors de l'authentification comme avec les empreintes digitales, donc pas de problème d'hygiène.

La méthode est encore trop récente pour être correctement évaluée. Le scanner est relativement encombrant comparé aux capteurs d'empreintes digitales.



**Figure 1.07 :** *Scan de réseau veineux*

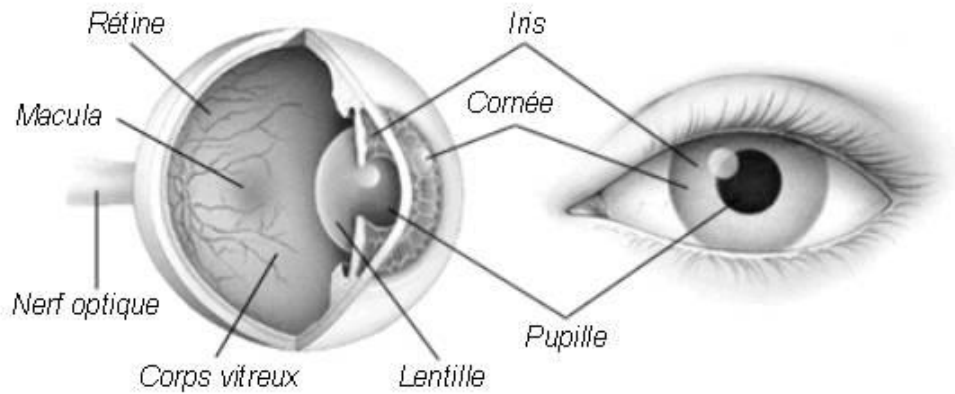
### ***1.4.4 Examen de l'œil***

#### 1.4.4.1 Rétine

La lecture des caractéristiques de la rétine est une technologie utilisée pour des applications de sécurité très élevée : par exemple, des systèmes de balayage de rétine ont été employés dans des applications militaires ou nucléaires.

Les caractéristiques de la rétine sont liées à la configuration géométrique des vaisseaux sanguins. La technologie utilise du matériel spécialisé et un rayon illumine le fond de l'œil.

Les systèmes identifient jusqu'à cent quatre-vingt-douze points de repère. Quelques risques pour la santé ont été révélés et limitent l'utilisation de cette technique à des locaux de haute sensibilité.



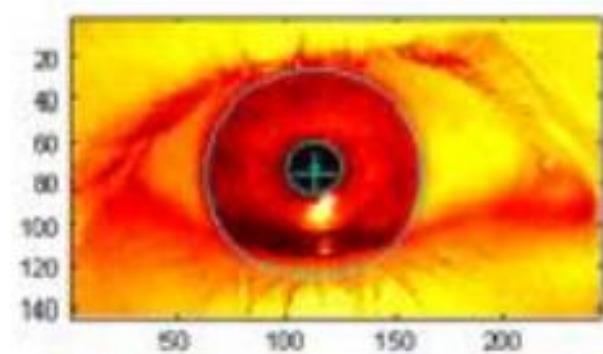
**Figure 1.08 : Coupe de l'œil**

L'avantage de ce système est qu'il est très résistant à la fraude, difficile et lente à imiter. Les caractères de la rétine sont très uniques même chez les vrais jumeaux ce qui rend ce système très fiable. La cartographie de la rétine est la même tout au long de la vie, sauf si la personne est atteinte d'une maladie spécifique.

On n'utilise pas très souvent ce système vu ses inconvénients. Il nécessite de placer ses yeux à très faible distance du capteur, le système intrusif est psychologiquement mal accepté. Ce système est difficile à utiliser en cas de contrôle de plusieurs personnes, car ceci impliquerait un temps important [7]. Sans parler du coût de déploiement avec l'installation qui est très délicate.

#### 1.4.4.2 Iris

L'iris est la région, sous forme d'anneau, situé entre la pupille et le blanc de l'œil. L'iris a une structure extraordinaire et offre de nombreuses caractéristiques de texture qui est unique pour chaque individu. Cela a donné lieu à une technologie d'identification fiable et extrêmement précise.



**Figure 1.09 : Détection de l'iris**

Avec les algorithmes utilisés dans la reconnaissance de l'iris, la planète tout entière pourrait être inscrite dans une base de données de l'iris avec peu d'erreurs d'identification.

Le capteur utilisé dans ce système est une simple caméra standard, mais l'œil doit être à une distance fixe assez proche du dispositif de capture, ce qui oblige l'implication de l'individu, limitant ainsi son utilisation.

#### ***1.4.5 Signature***

Il existe deux approches pour vérifier la signature : statique et dynamique.

Au niveau statique, la signature est normalisée à une taille connue et ensuite celui-ci sera décomposé en éléments simples. La forme et les relations de ses éléments sont utilisées comme caractéristiques d'identification.

Pourtant, la vérification dynamique de la signature analyse la façon dont un utilisateur signe son nom. Les caractéristiques dynamiques de la signature comme l'accélération, la vitesse et la précision sont aussi importantes que la forme géométrique de la signature.



**Figure 1.10 : Signature**

L'inconvénient de ce système est que la signature est une biométrie comportementale, elle évolue dans le temps et est influencée par les conditions physiques et émotionnelles de la personne [2]. Ce système fonctionne avec un capteur et un crayon lecteur ou stylo comme ceux présentés dans la figure 1.11.



**Figure 1.11 : Capteur de reconnaissance de signature**

Le capteur utilisé est une table digitalisée électromagnétique, de dimension 20cmx20cm. Ce capteur est relié à un PC, qui commande une ouverture de porte, l'accès à une base de données, etc. Tout mouvement du stylo en écriture est pris en compte (vitesse, rythme du stylo, accélération, pression, distance de suspension entre deux lettres, etc.), mais aussi jusqu'à environ 2 cm au-dessus de la tablette [7].

#### **1.4.6 Reconnaissance vocale**

##### **1.4.6.1 Définition**

La reconnaissance vocale ou ASR est une technologie de transcription d'un système « phonatoire organique » exploitable par une machine.

La reconnaissance vocale fait partie des domaines qu'il faut encore développer. Cette reconnaissance a des failles, car elle dépend de plusieurs paramètres comme la qualité du micro, le fait qu'il ne doit pas y avoir de sons parasites lors de l'analyse ou des bruits de fond. Celle-ci est souvent utilisée pour les transactions par téléphone pour minimiser les fraudes, mais elle n'est pas très efficace. L'un de ses avantages réside dans le fait que l'utilisateur n'est pas en contact direct avec l'appareil : son cerveau ne la perçoit pas comme intrusive.

La reconnaissance vocale couplée à des méthodes de synthèse vocale, de commande vocale, d'identification vocale, et de compréhension forment un élément d'une interface homme-machine idéale [8].

Selon certains sondages elle reste quand même une des méthodes les plus utilisées. La reconnaissance vocale serait utilisée à 32 % par les utilisateurs ensuite en second viendrait la reconnaissance par empreintes digitales à 27 % puis la reconnaissance faciale à 20 % et enfin les reconnaissances de la main et de l'iris à respectivement 12 et 10 %.

Ce système est en pleine croissance, car elle ne nécessite qu'un microphone pour le capteur. Seulement, le système de vérification peut être influencé par la mauvaise qualité du son ou par les bruits ambiants.

##### **1.4.6.2 Principe de fonctionnement**

De tous les traits humains utilisés dans la biométrie, la voix est celle que les humains apprennent à reconnaître dès leur plus jeune âge. Les systèmes de reconnaissance de locuteur peuvent être divisés en deux catégories : les systèmes dépendant du texte prononcé et les systèmes indépendants du texte. Dans le premier cas, l'utilisateur est tenu d'utiliser un texte (un mot ou une phrase) fixe prédéterminé

au cours des séances d'apprentissage et de reconnaissance. Alors que, pour un système indépendant du texte, le locuteur parle librement sans texte prédéfini.

Dans un système de reconnaissance de locuteur, après avoir acquis le signal, on le décompose en plusieurs canaux de fréquences passe-bande. Ainsi, on extrait les caractéristiques importantes du signal vocal dans chaque bande.

Les coefficients Cepstraux sont les caractéristiques les plus utilisées. Ils sont obtenus par le logarithme de la transformée de Fourier du signal vocal dans chaque bande. Et au final, la mise en correspondance des coefficients Cepstraux permet de reconnaître la voix. Dans cette étape, généralement on fait appel à des approches fondées sur les modèles de Markov cachés, la quantification vectorielle, ou la déformation temps dynamique.

On peut voir dans cette figure 1.12 le spectre d'un signal vocal capturé par un microphone.



**Figure 1.12 :** *Spectre d'un signal vocal*

Les systèmes d'identification de la voix sont basés sur les caractéristiques de voix, uniques pour chaque individu. Ces caractéristiques de la parole sont constituées par une combinaison des facteurs comportementaux (vitesse, rythme, etc.) et physiologiques (tonalité, âge, sexe, fréquence, accent, harmonie, etc.).

#### ***1.4.7 Reconnaissance faciale***

De même que la reconnaissance vocale, la reconnaissance faciale est une technique de différenciation d'individu qui n'est pas encore très optimale. Elle est actuellement expérimentale et a pour base la photographie d'une personne permettant de l'identifier. Cette technique est utilisée par exemple via une webcam qui capture l'image du visage présentée et qui, par le biais d'un logiciel, va la numériser pour créer une base de données.

La plupart des algorithmes de reconnaissance de visage cherchent à extraire un code descriptif réduit du visage (entre 10 et 500 paramètres), puis à rattacher ce code à une classe statistique correspondant à un individu [5].

La méthode employée consiste à localiser les informations utiles pour le système, c'est-à-dire localiser seulement le visage, extraire des caractéristiques à partir de ce visage, comparer ces caractéristiques avec d'autres qui existent dans la base de données images. Au cas où ces caractéristiques se correspondent alors le visage est identifié ou reconnu [2].

La réduction du champ de la recherche (qui est le rôle de la procédure de classification) limite le nombre des visages à parcourir en choisissant seulement ceux qui sont susceptibles de correspondre au visage cherché, c'est-à-dire ceux qui ont des caractéristiques qui répondent au critère de classification puis on procédera à l'identification.

## **1.5 Conclusion**

Ce chapitre montre l'importance de la biométrie dans les systèmes de sécurité et cela depuis quelques années. En général, tous les systèmes biométriques comportent deux processus principaux, l'apprentissage et la reconnaissance. On a aussi vu quelques applications biométriques les plus connues et les plus utilisées, chacune présente des avantages et des inconvénients. Malgré ses limites, la reconnaissance faciale a plusieurs avantages sur d'autres technologies biométriques: c'est naturel, non intrusif, et facile à utiliser. On va voir plus de détails sur la reconnaissance faciale dans le chapitre suivant.



## **CHAPITRE 2**

### **RECONNAISSANCE FACIALE**

#### **2.1 Généralités**

##### ***2.1.1 Introduction***

La reconnaissance de visages est la technique la plus commune et populaire. Elle reste la plus acceptable puisqu'elle correspond à ce que les humains utilisent dans l'interaction visuelle ; et par rapport aux autres méthodes, la reconnaissance du visage s'avère plus avantageuse, d'une part c'est une méthode non intrusive, c'est-à-dire qu'elle n'exige pas la coopération du sujet (en observant les individus à distance), et d'autre part les capteurs utilisés sont peu coûteux (une simple caméra) contrairement à l'empreinte digitale et l'iris où le sujet devra être très proche du capteur et devra coopérer pour l'acquisition de l'image sans oublier le coût de l'équipement nécessaire pour l'acquisition (équipement spécial coûteux) [9].

Mais la reconnaissance des visages est une biométrie relativement peu sûre, car certains problèmes diminuent sa fiabilité. Sur ce fait, le signal acquis est sujet à des variations beaucoup plus élevées que d'autres caractéristiques, comme la variation de l'éclairage, le changement de la position du visage, la présence ou l'absence de lunettes et autres.

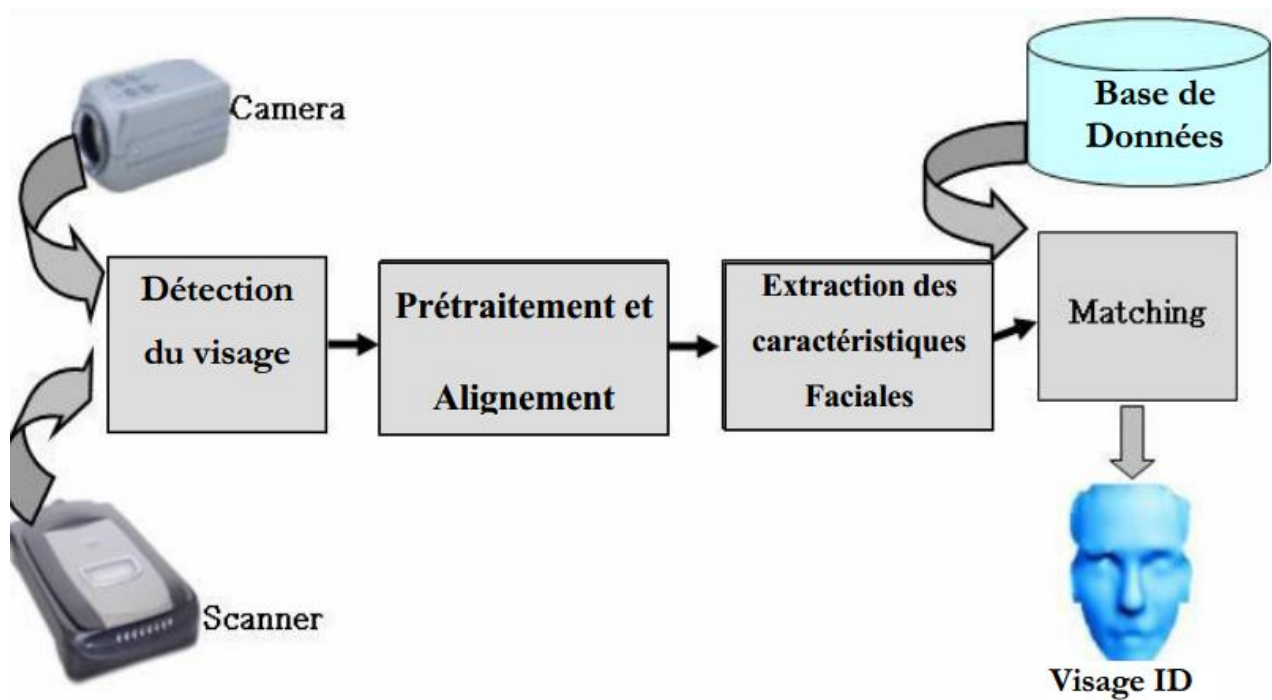
##### ***2.1.2 Définition***

La reconnaissance des visages est une technique d'identification des individus basée sur leurs caractéristiques faciales. L'identification du visage en question revient à trouver une ou plusieurs images du visage correspondant dans une base de données.

##### ***2.1.3 Processus de reconnaissance faciale***

Les méthodes de reconnaissance du visage diffèrent l'une par rapport à l'autre, cependant le processus est le même, en effet la reconnaissance automatique des visages humains peut être divisée en deux types d'applications : l'identification et la vérification. Dans le problème d'identification, le visage à reconnaître est comparé avec les visages dans la base de données. Cependant dans le problème de vérification, le système doit confirmer si le visage à l'entrée appartient à la base de visage, dans le cas contraire, ce visage sera rejeté.

Quel que soit le type d'application, le processus de reconnaissance faciale est le même. Ce processus est généralement constitué des quatre phases suivantes, voir figure 2.01 [10].



**Figure 2.01 :** *Processus général de reconnaissance faciale*

#### 2.1.3.1 Détection

C'est la première phase dans le processus de reconnaissance automatique des visages, la détection joue un rôle très important et a une très grande influence sur la fiabilité et la performance de tout le système de reconnaissance. Etant donné une image fixe ou dans une séquence vidéo, la détection du visage dans cette scène doit être capable de localiser le visage dans sa totalité, quelles que soient sa position, l'échelle, l'orientation, les conditions d'éclairage et les expressions faciales, etc.

La détection peut se baser sur différents critères (c'est-à-dire la couleur de la peau, l'émotion, la forme de la tête, etc.).

#### 2.1.3.2 Prétraitement

C'est une phase importante, son objectif est l'élimination des bruits dans l'image et sa normalisation selon une taille, un contraste, une distribution des pixels et une intensité adéquate, correspondants aux exigences du concepteur.

#### 2.1.3.3 Alignement du visage

L'objectif de l'alignement est la localisation des caractéristiques faciales telles que les yeux, le nez et la bouche pour pouvoir par la suite mettre le visage à l'échelle et dans la zone adéquate du traitement.

#### 2.1.3.4 Extraction des caractéristiques

L'objectif de cette phase est la définition et la génération d'un ensemble de caractéristiques les plus représentatives du visage. Cette phase réduit énormément la taille des données à traiter et peut réduire le temps de reconnaissance d'une manière très significative.

#### 2.1.3.5 Reconnaissance

L'objectif de cette étape est la reconnaissance du visage en se basant uniquement sur ces caractéristiques principales, dans cette étape une procédure du matching est exécutée pour déterminer les visages les plus proches et qui ont une correspondance très proche à celui en entrée.

#### 2.1.3.6 Classification

Basé sur le résultat du matching, le classificateur doit apprendre une fonction de décision afin d'implémenter la décision sur la classification finale du visage. Dans cette étape le système est censé trouver la personne dans la base des visages correspondant aux caractéristiques mentionnées, ou bien rejeter le visage dans le cas contraire.

### 2.2 Approches de la détection de visage

Il existe plusieurs méthodes pour détecter les visages. Voici les quatre différents types de détection de visage.

#### 2.2.1 Méthodes basées sur l'apparence

Cette approche a pour objectif de déterminer les caractéristiques significatives des visages et des non-visages à partir de techniques d'analyse statistique et d'apprentissage organisés par le biais de modèles de distribution ou par une fonction discriminante.

Ces approches appliquent généralement des techniques d'apprentissage automatique. Ainsi, les modèles sont appris à partir d'un ensemble d'images représentatives de la variabilité de l'aspect facial. Ces modèles sont alors employés pour la détection. L'idée principale de ces méthodes est de considérer que le problème de la détection de visage est un problème de classification (visage, non-visage). Une des approches les plus connues de détection de visage est l'eigenfaces. Elle consiste à projeter l'image dans un espace et à calculer la distance euclidienne entre l'image et sa projection. En effet, en codant l'image dans un espace, on dégrade l'information contenue dans l'image, puis on calcule la perte d'informations entre l'image et sa projection. Si cette perte d'informations est grande (évaluée à partir de la distance, que l'on compare à un seuil fixé a priori), l'image n'est pas correctement représentée dans l'espace : elle ne contient pas de visage [2] [11].

### ***2.2.2 Méthodes basées sur les connaissances acquises***

Ces méthodes sont fondées sur des règles qui représentent les composants principaux et représentatifs des visages humains. Les règles sont généralement constituées à partir de la relation entre les caractéristiques du visage. Elles s'intéressent aux parties caractéristiques du visage comme le nez, la bouche et les yeux. Ces méthodes sont conçues principalement pour la localisation de visage.

Par exemple, les visages sur les images ont souvent deux yeux qui sont symétriques, un nez et une bouche. La relation entre ces membres peut être représentée par la distance entre ces derniers et leur position.

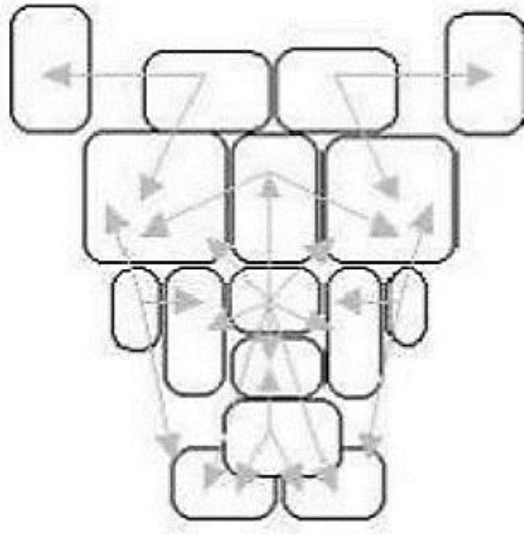
Un problème se pose avec cette approche, en l'occurrence la difficulté de traduire les connaissances relatives aux visages humains en règle bien définies, ce qui peut provoquer des erreurs de détection et rendre le système peu fiable [2] [11].

### ***2.2.3 Méthodes basées sur l'appariement de gabarits « Template-matching »***

La détection de visage entier ou de parties de visage se fait à travers un apprentissage d'exemples standards de visages. La corrélation entre les images d'entrées et les exemples enregistrés est calculée et utilisée pour la décision.

Les templates peuvent être définis soit « manuellement », soit paramétrés à l'aide de fonctions. L'idée est de calculer la corrélation entre l'image candidate et le template. Ces méthodes rencontrent encore quelques problèmes de robustesses liés aux variations de lumière, d'échelle, etc. Sinhn [12] utilise un ensemble d'invariants décrivant le modèle du visage. Afin de déterminer les invariants aux changements de luminosité permettant de caractériser les différentes parties du visage (telles que les yeux, les joues, et le front), cet algorithme calcule les rapports (par exemple, la région 1 est-elle plus claire ou plus sombre que la région 2 ?) [2].

Cette technique est utilisée pour classer des objets, elle est très intéressante pour la détection de visage de par sa facilité d'application. Le principe de cette méthode est basé sur une comparaison effectuée entre une image quelconque et un modèle prédéfini, dont le but est de calculer la corrélation pour aboutir à une décision par oui ou par non. La correspondance est faite pixel par pixel [11].



**Figure 2.02 :** *Modèle de visage composé de 16 régions (les rectangles) associées à 23 relations (flèches).*

La figure 2.02 montre un modèle prédéfini correspondant à 23 relations. Ces relations prédéfinies sont classifiées en 11 relations essentielles et 12 relations de confirmations. Chaque flèche représente une relation entre deux régions. Une relation est vérifiée si le rapport entre les deux régions qui lui correspondent dépasse un seuil. Le visage est localisé si le nombre de relations essentielles et de confirmation dépasse lui aussi un seuil.

#### **2.2.4 Méthodes basées sur les caractéristiques du visage**

Ces approches sont utilisées principalement pour la localisation de visage. Les algorithmes développés visent à trouver les caractéristiques structurales existantes même si la pose, le point de vue ou les conditions d'éclairage changent. Puis ils emploient ces caractéristiques invariables pour localiser les visages. Nous pouvons citer deux familles de méthodes appartenant à cette approche : les méthodes basées sur la couleur de la peau et les méthodes basées sur les caractéristiques de visage.

On peut voir dans la figure 2.03 la détection de visage basée sur la couleur de la peau.

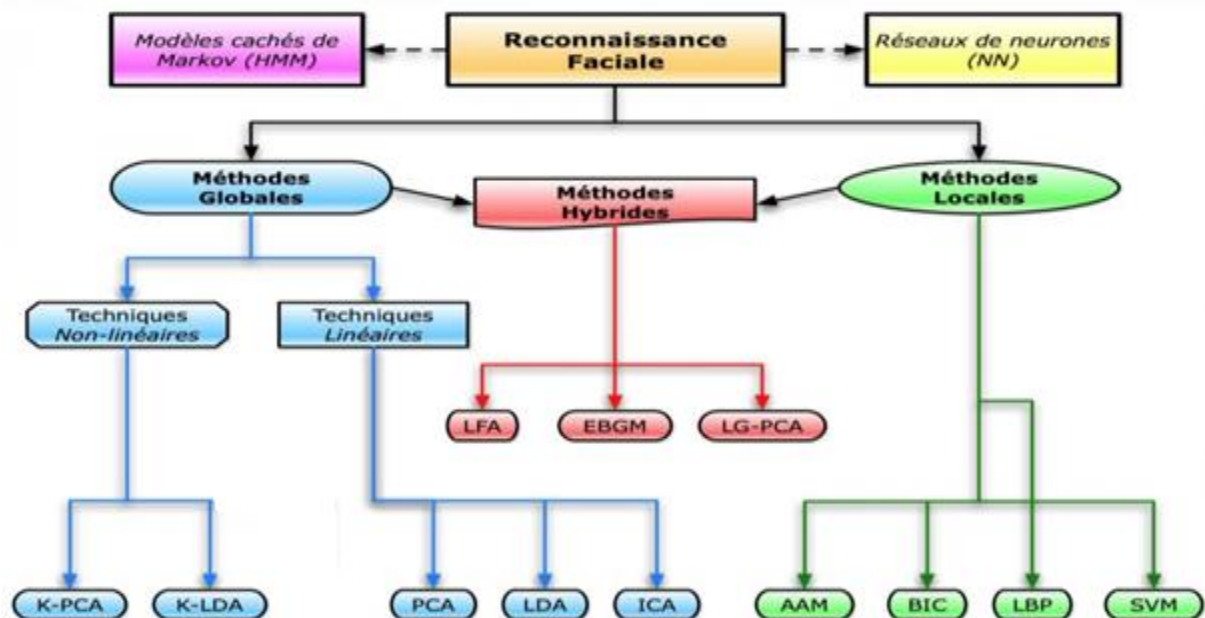


**Figure 2.03 :** *la détection de la couleur de peau.*

Le problème que rencontre cette approche est que la qualité des images peut être sévèrement diminuée à cause de l'illumination, le bruit ou l'occlusion.

## 2.3 Algorithme de reconnaissance faciale

Les méthodes de reconnaissance des visages peuvent être classées en trois grandes approches comme la figure 2.04 nous le montre : une approche globale dans laquelle on analyse le visage (l'image pixellisée du visage) dans son entier ; une approche locale basée sur un modèle, dans laquelle le système essaie de détecter, regrouper et reconnaître les différents éléments constitutifs du visage tels que le nez, les yeux et la bouche ; enfin, il existe des méthodes hybrides qui combinent les deux approches précédentes [2].



**Figure 2.03 :** Classification des algorithmes principaux utilisés en reconnaissance faciale

### 2.3.1 Méthodes globales

La méthode globale aussi appelée méthode holistique est un principe qui utilise toute la surface du visage comme source d'information à l'entrée de l'algorithme, sans tenir compte des caractéristiques locales du visage telles que les yeux, le nez, etc. Ces techniques sont très réussies et bien étudiées, ainsi elles offrent de très bonnes performances. Mais le problème de stockage des informations extraites lors de la phase d'apprentissage reste le problème majeur de cette méthode [11] [13].

Généralement, ces méthodes sont basées sur des techniques d'analyse statistique bien connues.

Les méthodes globales peuvent être divisées en deux grandes catégories à savoir les méthodes linéaires et les méthodes non linéaires [14].

Les techniques linéaires projettent linéairement les données d'un espace de grande dimension (par exemple, l'espace de l'image originale) sur un sous-espace de dimension inférieure. Malheureusement, ces techniques sont incapables de préserver les variations non convexes (géométriques au sens mathématique du terme) des variétés de visages afin de différencier des individus. Dans un sous-espace linéaire, les distances euclidiennes et plus généralement les distances de Mahalanobis, qui sont normalement utilisées pour comparer des vecteurs de données, ne permettent pas une bonne classification entre les classes de formes « visage » et « non-visage » et entre les individus eux-mêmes. Ce facteur crucial limite le pouvoir des techniques linéaires pour obtenir une détection et une reconnaissance faciale très précises.

La technique linéaire la plus connue et sans aucun doute l'analyse en composantes principales (ACP), également appelée transformée de Karhunen-Loeve. L'ACP fut d'abord utilisé afin de représenter efficacement des images de visages humains. C'est en 1991 que cette technique a été reprise dans le cadre plus spécifique de la reconnaissance faciale sous le nom de méthode des eigenfaces [2].

Il existe d'autres techniques également construites à partir de décompositions linéaires comme l'Analyse Discriminante linéaire (LDA en anglais) ou encore l'Analyse en Composantes Indépendantes (ICA en anglais) [2] [9].

Bien que ces méthodes globales linéaires basées sur l'apparence évitent l'instabilité des toutes premières méthodes géométriques qui ont été mises au point, elles ne sont pas assez précises pour décrire les subtilités des variétés (géométriques) présentes dans l'espace de l'image originale. Ceci est dû à leurs limitations à gérer la non-linéarité en reconnaissance faciale : les déformations de variétés non linéaires peuvent être lissées et les concavités peuvent être remplies, causant des conséquences défavorables.

Afin de pouvoir traiter ce problème de non-linéarité en reconnaissance faciale, de telles méthodes linéaires ont été étendues à des techniques non linéaires basées sur la notion mathématique de noyau (kernel) comme le Kernel ACP et Kernel ADL. Ici, une projection non linéaire (réduction de dimension) de l'espace de l'image sur l'espace de caractéristiques (feature space) est effectuée ; les variétés présentes dans l'espace de caractéristiques résultantes deviennent simples, de même que les subtilités des variétés qui sont préservées. Bien que les méthodes basées sur le noyau peuvent atteindre une bonne performance sur les données d'apprentissage, il ne peut pas en être de même

pour de nouvelles données en raison de leur plus grande flexibilité ; contrairement aux méthodes linéaires [2].

### 2.3.2 Méthodes locales

Les méthodes locales sont aussi appelées méthode à traits, géométriques, à caractéristiques locales, ou analytiques.

L'analyse du visage humain est donnée par la description individuelle de ses parties et de leurs relations. Ce modèle correspond à la manière avec laquelle l'être humain perçoit le visage, c'est à dire, à nos notions de traits de visage et de parties comme les yeux, le nez, la bouche, etc. [14].

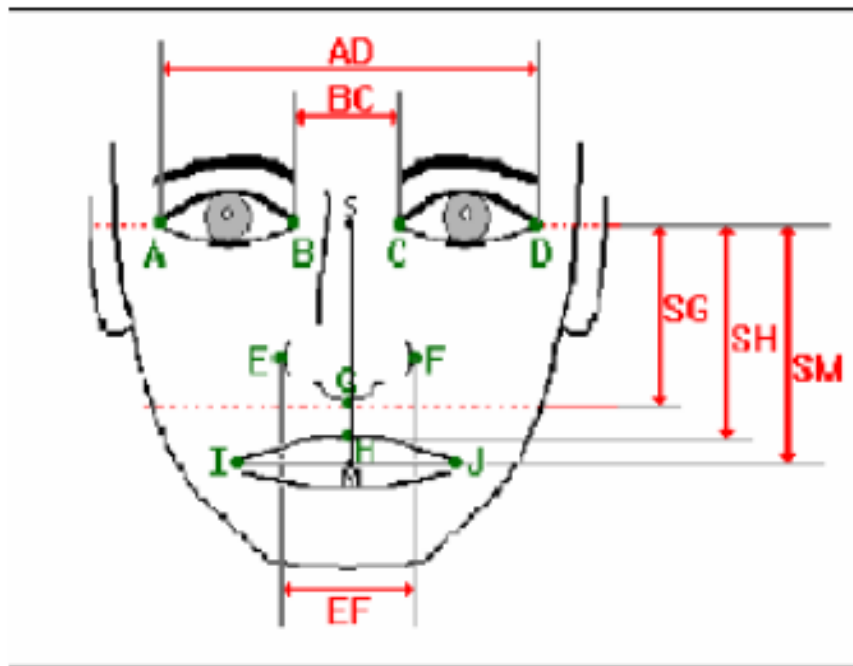
Catégorie	Traits
<b>Yeux</b>	Forme, couleur, distance entre les yeux
<b>Bouche</b>	Gabarit, largeur, longueur, superficie de la bouche ouverte
<b>Sourcils</b>	Séparation, épaisseur
<b>Cheveux</b>	Intensité, forme, couleur
<b>Lèvres</b>	Largeur, forme, couleur
<b>Joues</b>	Intensité
<b>Nez</b>	Longueur, largeur
<b>Distances</b>	d(yeux, centre du nez), d(menton, bouche)...
<b>Rapports</b>	d(centre de la face, bouche)/d(menton, bouche)...

**Tableau 2.01 :** Traits utilisés pour identifier un visage par la méthode locale.

L'avantage de ces méthodes est qu'elles prennent en compte la particularité du visage en tant que forme naturelle à reconnaître, en exploitant les résultats de la recherche suivant la neuropsychologie et la psychologie cognitive sur le système visuel humain. Ce qui leur permet de modéliser plus facilement les variations de pose, d'éclairage et d'expression par rapport aux méthodes globales. Ces méthodes éprouvent des difficultés quand il s'agit de prendre en considération plusieurs vues des visages ; Ainsi le manque de précision dans la phase « extraction » des points constitue leur inconvénient majeur. Toutefois, elles sont plus lourdes à utiliser puisqu'il faut souvent placer manuellement un assez grand nombre de points sur le visage alors que les méthodes globales ne nécessitent de connaître que la position des yeux afin de normaliser les images, ce qui peut être fait automatiquement et de manière assez fiable par un algorithme de détection [11] [15].



La figure 2.04 montre un exemple de mesure de distance entre des points caractéristiques du visage. Il existe plusieurs façons de prendre ces distances, mais en générale, le principe reste le même.



**Figure 2.04 :** *Distances entre points caractéristiques*

Précisément, ces méthodes extraient les caractéristiques locales du visage, puis utilisent leur géométrie et/ou l'apparence comme données d'entrée du classificateur. On peut distinguer deux pratiques différentes [9] :

- La première repose sur l'extraction de régions entières du visage, elle est souvent implémentée avec une approche globale de reconnaissance de visage, par exemple l'eigenfaces modulaire.
- La deuxième pratique extrait des points particuliers des différentes régions caractéristiques du visage, tels que les coins des yeux, de la bouche et du nez. Parmi ces approches on peut citer : les Modèles de Markov Cachés (HMM), l'Algorithme Elastic Bunch Graph Matching (EBGM), Eigen Object (EO), l'appariement de gabarits.

### **2.3.3 Méthodes hybrides**

Les méthodes hybrides sont des approches qui combinent les caractéristiques holistiques avec les caractéristiques d'apparences locales afin d'améliorer les performances de la reconnaissance du visage. En effet, les caractéristiques locales et les caractéristiques globales ont des propriétés tout à fait différentes. Elles permettent ainsi d'hériter des avantages des deux méthodes.

On augmente ainsi la stabilité de la performance de reconnaissance lors de changements de pose, d'éclairage et d'expressions faciales [12] [15].

Le tableau 2.02 récapitule qualitativement la différence entre les deux types de caractéristiques.

<b>Facteurs de variations</b>	<b>Caractéristiques locales</b>	<b>Caractéristiques globales</b>
<b>Illuminations</b>	Très sensible	Sensible
<b>Expressions</b>	Non sensible	Sensible
<b>Pose</b>	Sensible	Très sensible
<b>Bruit</b>	Très sensible	Sensible
<b>Occlusion</b>	Pas sensible	Très sensible

**Tableau 2.02 :** *Comparaison des méthodes basées sur les caractéristiques locales ou globales.*

Nous pouvons voir que les caractéristiques locales et globales réagissent différemment aux facteurs de variation. Par exemple, les changements d'illumination peuvent avoir plus d'influence sur les caractéristiques locales, tandis que les changements d'expression ont plus d'impact sur les caractéristiques holistiques. Ainsi, les méthodes hybrides peuvent constituer une approche efficace pour réduire la complexité des classificateurs et améliorer leur capacité de généralisation [13].

L'analyse de caractéristiques locales (LFA hybride) et les caractéristiques extraites par ondelettes de Gabor (comme l'EBGM), sont des algorithmes hybrides typiques. Il y a aussi des algorithmes plus récents comme le Log Gabor ACP (LG-ACP) qui effectue une convolution avec des ondelettes de Gabor orientées autour de certains points caractéristiques du visage [2].

## 2.4 Conclusion

Dans ce chapitre, on vient d'approfondir l'état de l'art de la reconnaissance faciale. Suivant un processus commun, toutes les reconnaissances de visage passent de la détection à la reconnaissance. Diverses approches de détection de visage ont été présentées ainsi que des algorithmes de reconnaissance faciale. L'ACP faisant partie des méthodes de reconnaissance faciale globale, possède une variante en méthode locale : l'ACP modulaire ; et c'est ce qui a retenu notre attention. On présentera ainsi l'ACP dans le chapitre suivant.

## **CHAPITRE 3**

### **ANALYSE EN COMPOSANTES PRINCIPALES**

#### **3.1 Présentation**

##### ***3.1.1 Introduction et définition***

L'analyse en composantes principales (ACP), en anglais Principal Components Analysis (PCA), est une des plus anciennes techniques d'analyse statistique. Elle a été développée pour l'étude d'échantillons d'individus caractérisés par plusieurs facteurs. La méthode est donc adaptée à l'analyse de données multidimensionnelles : en effet, l'étude séparée de chaque facteur ne suffit pas en général, car elle ne permet pas de détecter d'éventuelles dépendances entre facteurs [5].

L'Analyse en Composantes Principales, aussi connues sous le nom de transformée de Karhunen–Loeve est une méthode très utilisée en statistique. Sa principale idée est de réduire la dimension d'un jeu de données tout en gardant un maximum d'informations. Cela est réalisé grâce à une projection qui maximise la variance tout en minimisant l'erreur quadratique moyenne de la reconstruction.

Selon [16], l'ACP est aussi l'application statistique la plus fréquente de la décomposition en valeurs singulières. On l'utilise pour étudier une matrice de données dans laquelle on recueille les valeurs de plusieurs variables quantitatives pour un ensemble d'observations. Ces variables sont, en général, corrélées entre elles et il est donc difficile d'apprécier l'effet d'une variable indépendamment des autres. Le but de l'analyse est de remplacer ces variables d'origine par un nouvel ensemble de variables orthogonales appelées « composantes principales ».

##### ***3.1.2 Brève histoire de l'ACP***

Conçue par Karl Pearson en 1901, intégrée à la statistique mathématique par Harold Hotelling en 1933, l'analyse en composantes principales n'est vraiment utilisée que depuis la large diffusion des moyens de calcul informatique.

La technique d'analyse en composantes principales peut être présentée suivant divers points de vue. Pour le statisticien classique, il s'agit de la recherche des axes principaux de l'ellipsoïde d'une distribution normale multidimensionnelle, ces axes étant estimés à partir d'un échantillon. C'est la présentation initiale de Hotelling (1933), puis celle des manuels classiques d'analyse multivariée.

Pour le factorieliste classique, il s'agit d'un cas particulier de la méthode d'analyse factorielle des psychométriciens (cas de variances spécifiques nuls ou égaux).

Enfin, du point de vue plus récent des analystes de données, il s'agit d'une technique de représentation des données, ayant un caractère optimal selon certains critères algébriques et géométriques, et que l'on utilise en général sans référence à des hypothèses de nature statistique ni à un modèle particulier.

Ce point de vue, fort répandu actuellement est peut-être le plus ancien. C'est celui qui avait été adopté par Pearson (1901). Bien entendu, il ne s'agissait pas de l'analyse en composantes principales telle que nous la présentons, mais les idées essentielles de la méthode étaient déjà présentées par cet auteur.

L'analyse en composantes principales présente de nombreuses variantes selon les transformations apportées au tableau de données : le nuage des points-individus peut être centré ou non, réduit ou non. Parmi ces variantes, l'analyse en composantes principales normée (nuage centré-réduit) est certainement la plus utilisée et c'est celle-ci que nous choisissons pour présenter les principes de l'analyse [17].

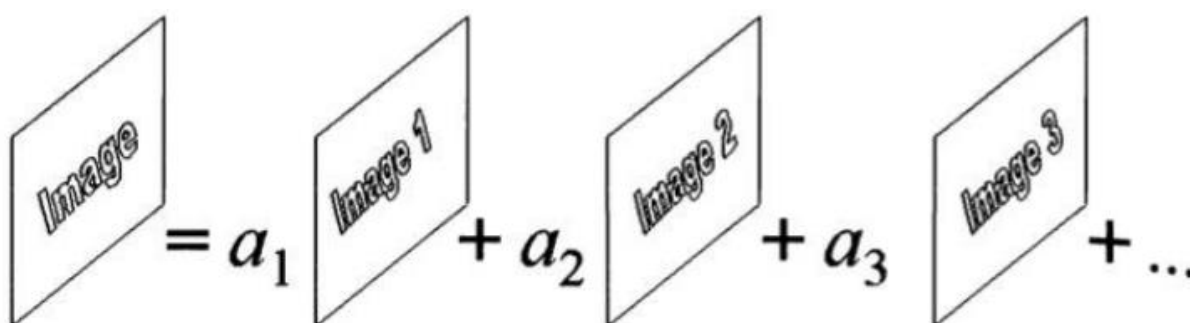
### **3.2 Transformation unitaire : Images et ACP**

Une transformation unitaire est un cas particulier de transformation linéaire qui permet de décomposer une image en un ensemble de fonctions simples à deux dimensions, appelées « fonctions de base ». De façon plus concrète, une transformation unitaire revient à décomposer une image en une base d'images élémentaires, représentées sur la figure 3.01. A chaque image de la base correspond un poids qui indique l'importance de cette image dans la reconstruction de l'image originale [16]. Ces poids correspondent à l'amplitude des images ou fonctions de base et sont appelés les « coefficients » de la transformation. Une transformation unitaire se doit de respecter les propriétés suivantes :

- Elle est exactement réversible : il existe une transformation inverse qui permet d'aller de l'image transformée à l'image d'origine.
- Elle est composée de bases orthogonales : les fonctions ou images de base sont indépendantes les unes des autres.
- Elle préserve l'énergie (ou variance) de l'image : l'énergie de l'image transformée est égale à l'énergie de l'image originale.
- Les premières fonctions ou images de base concentrent la plus grande partie de l'énergie, ou inertie, de l'image.

- La différence des écarts au carré entre deux images transformées est égale à la différence des écarts au carré entre les deux images initiales.

Puisque la plus grande partie de l'énergie de l'image est regroupée dans les premières images de la base, beaucoup d'images de la base ne contiennent qu'une petite partie de l'énergie de l'image. Ces images peuvent donc être éliminées sans entraîner une réelle perte d'informations. Une image peut ainsi être synthétisée à partir des premières images de la base.



**Figure 3.01 :** Décomposition d'une image en une somme pondérée d'images élémentaires.

### 3.2.1 Etape 1 : Numériser et vectoriser des images

Chaque image est représentée par une matrice de pixels. Ici, on utilise un code binaire pour représenter les images par des matrices de dimensions 3x3 comme sur la figure 3.02.



**Figure 3.02 :** Quatre images binaires

$$Y_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} Y_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} Y_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} Y_4 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.01)$$

Chaque élément de ces matrices représente le niveau de gris du pixel correspondant : 0 pour blanc et 1 pour noir. La vectorisation est ensuite effectuée à l'aide de l'opérateur  $\text{vec}$ .

En mathématiques, en particulier dans l'algèbre linéaire et la théorie des matrices, la vectorisation d'une matrice est une transformation linéaire qui convertit la matrice dans un vecteur de colonne (Voir ANNEXE 1). Plus précisément, la vectorisation d'une matrice  $A(m \times n)$ , notée  $\text{vec}(A)$  est le vecteur colonne  $1 \times (m \times n)$  obtenue en associant les colonnes de la matrice  $A$  au-dessus d'une autre. Chaque image est donc maintenant représentée par un vecteur de dimension  $1 \times 9$ .

$$\begin{aligned}
x_1 &= \text{vec}(Y_1) = [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]^T \\
x_2 &= \text{vec}(Y_2) = [0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0]^T \\
x_3 &= \text{vec}(Y_3) = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1]^T \\
x_4 &= \text{vec}(Y_4) = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1]^T
\end{aligned} \tag{3.02}$$

L'ensemble des images est représenté par la matrice d'ordre 9x4 :

$$X = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \tag{3.03}$$

La k-ième colonne de cette matrice représente la k-ième image et la i-ème ligne représente la i-ème pixel.

### 3.2.2 Etape 2 : Calculer la matrice de produits croisés

La deuxième étape consiste à calculer la matrice de produits croisés des pixels :

$$X^T X = \begin{bmatrix} 3 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 3 \\ 1 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 \\ 2 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 2 \\ 1 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 & 0 & 2 & 1 & 2 & 1 \\ 3 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 3 \end{bmatrix} \tag{3.04}$$

Les éléments diagonaux de cette matrice représentent l'inertie des pixels. Plus un pixel diffère d'une image à l'autre, plus cette valeur est élevée. Les éléments non-diagonaux représentent la covariation entre deux pixels.

### 3.2.3 Etape 3 : Diagonaliser la matrice de produits croisés

L'étape suivante consiste à trouver les vecteurs et valeurs propres de la matrice  $XX^T$  :

$$XX^T = U\Lambda U^T \tag{3.05}$$

Après quelque étape de calculs, on trouve ainsi :

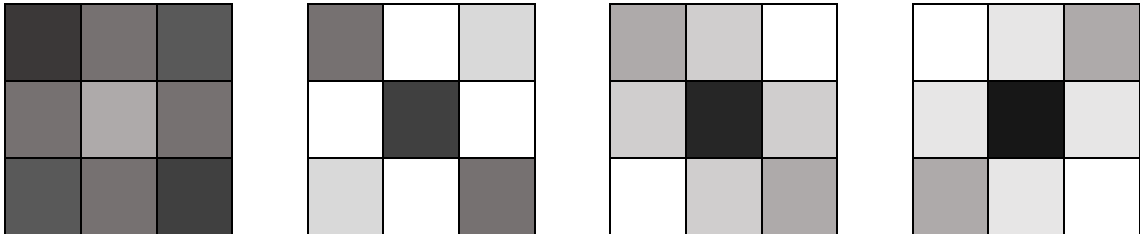
$$U = \begin{bmatrix} 0.42 & 0.32 & 0.18 & -0.43 \\ 0.30 & -0.38 & 0.11 & 0.06 \\ 0.34 & 0.13 & -0.57 & 0.19 \\ 0.30 & -0.38 & 0.11 & 0.06 \\ 0.21 & 0.43 & 0.48 & 0.74 \\ 0.30 & -0.38 & 0.11 & 0.06 \\ 0.34 & 0.13 & -0.57 & 0.19 \\ 0.30 & -0.38 & 0.11 & 0.06 \\ -0.42 & 0.32 & 0.18 & -0.43 \end{bmatrix} \quad (3.06)$$

Et

$$\Lambda = \begin{bmatrix} 13.09 & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.60 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.11 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.20 \end{bmatrix} \quad (3.07)$$

On peut ensuite former une image par la représentation matricielle de chaque vecteur propre. On obtient ainsi quatre images de base, ou images propres comme sur la figure 3.03 qui serviront à reconstruire les quatre images initiales :

$$\begin{aligned} U_1 &= \begin{bmatrix} 0.42 & 0.30 & 0.34 \\ 0.30 & 0.21 & 0.30 \\ 0.34 & 0.30 & 0.42 \end{bmatrix} & U_2 &= \begin{bmatrix} 0.32 & -0.38 & 0.13 \\ -0.38 & 0.43 & -0.38 \\ 0.13 & -0.38 & 0.32 \end{bmatrix} \\ U_3 &= \begin{bmatrix} 0.18 & 0.11 & -0.57 \\ 0.11 & 0.48 & 0.11 \\ -0.57 & 0.11 & 0.78 \end{bmatrix} & U_4 &= \begin{bmatrix} -0.43 & 0.06 & 0.19 \\ 0.06 & 0.74 & 0.06 \\ 0.19 & 0.06 & -0.43 \end{bmatrix} \end{aligned} \quad (3.08)$$



**Figure 3.03 :** Base d'images extraites des images présentées dans la figure 3.02

#### 3.2.4 Etape 4 : Calcul des projections des images

Les projections des images sur les images de la base sont obtenues en multipliant la transposée de la matrice X par la matrice de vecteurs propres U :

$$G = X^T U = \begin{bmatrix} 1.06 & 1.08 & 0.84 & -0.12 \\ 1.20 & -1.51 & 0.46 & 0.24 \\ 1.74 & 1.34 & -0.31 & 0.27 \\ 2.73 & -0.61 & -0.33 & -0.23 \end{bmatrix} \quad (3.09)$$

La première colonne de G représente les projections des quatre images sur le premier vecteur propre. Ces projections indiquent la quantité de l'image de base nécessaire à la reconstruction de chaque image initiale [16].

### 3.2.5 Etape 5 : Reconstruction des images initiales

Les vecteurs  $x_k$  peuvent être reconstruits comme une somme pondérée de vecteurs propres :

$$x_k = \sum_{l=1}^L g_{k,l} u_l \quad (3.10)$$

Les poids, notés  $g_{k,l}$ , sont les projections du vecteur  $x_k$  sur l-ième vecteur propre de  $XX^T$ . Les poids des quatre images sur le premier vecteur propre,  $g_1$ , sont donnés par la première ligne de la matrice G. Les poids sur le second vecteur propre,  $g_2$ , sont donnés par la deuxième ligne de la matrice G, etc. Pour les quatre images, on obtient :

$$\begin{aligned} x_1 &= 1.06 u_1 + 1.08 u_2 + 0.84 u_3 - 0.12 u_4 \\ x_2 &= 1.20 u_1 - 1.51 u_2 + 0.46 u_3 + 0.24 u_4 \\ x_3 &= 1.74 u_1 + 1.34 u_2 - 0.31 u_3 + 0.27 u_4 \\ x_4 &= 2.73 u_1 - 0.67 u_2 - 0.30 u_3 - 0.23 u_4 \end{aligned} \quad (3.11)$$

On peut vérifier, par exemple, que :

$$x_1 = 1.06 * \begin{bmatrix} .42 \\ .30 \\ .34 \\ .30 \\ .21 \\ .30 \\ .34 \\ .30 \\ .42 \end{bmatrix} + 1.08 * \begin{bmatrix} .32 \\ -.38 \\ .13 \\ -.38 \\ .43 \\ -.38 \\ .13 \\ -.38 \\ .32 \end{bmatrix} + .84 * \begin{bmatrix} .18 \\ .11 \\ -.57 \\ .11 \\ .48 \\ .11 \\ -.57 \\ .11 \\ .18 \end{bmatrix} - .12 * \begin{bmatrix} -.43 \\ .06 \\ .19 \\ .06 \\ .74 \\ .06 \\ .19 \\ .06 \\ -.43 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.12)$$

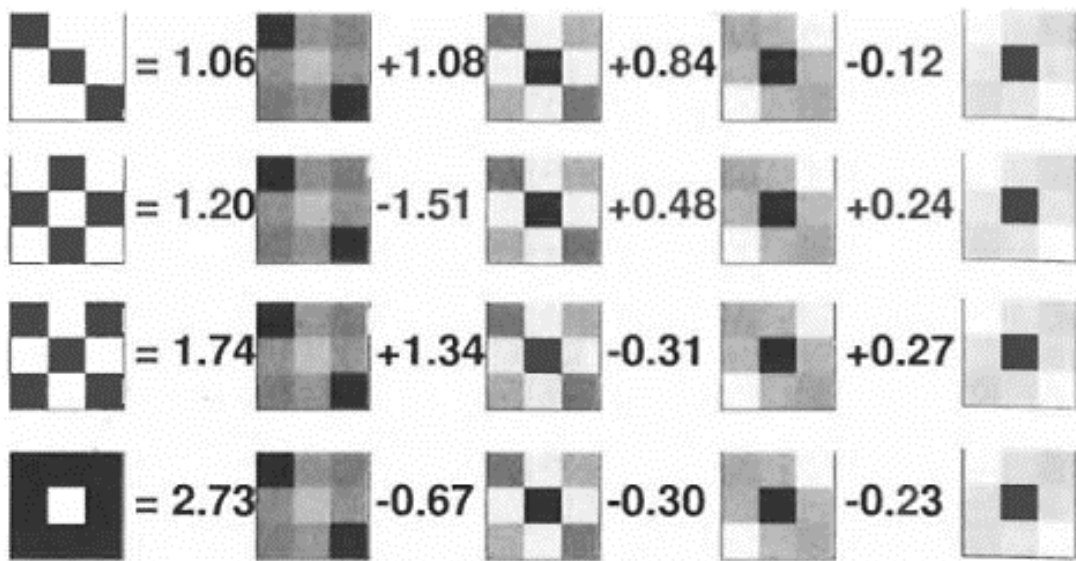
Les quatre vecteurs peuvent être reconstruits en une seule opération :

$$X = UG^T = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad (3.13)$$

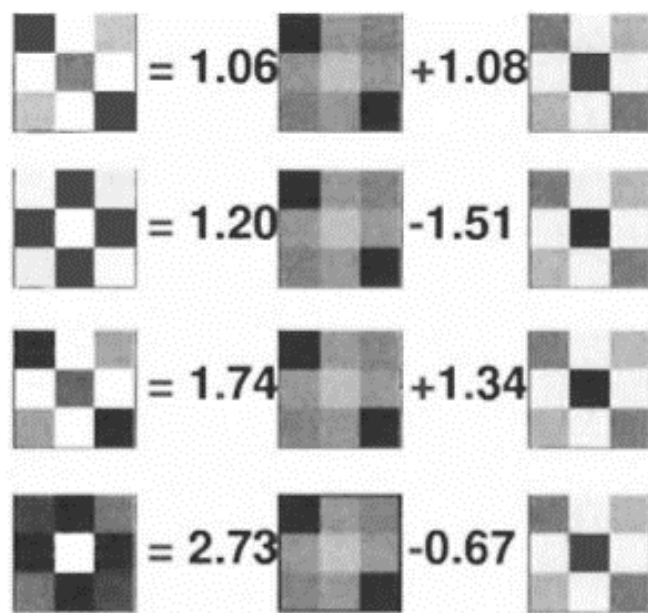


Comme l'illustre la figure 3.04, les images elles-mêmes peuvent ensuite être reconstruites en repliant les vecteurs  $x_k$ .

Les valeurs propres associées à une image de la base donnent l'inertie de cette image. L'importance de cette inertie peut être estimée en la divisant par la somme des valeurs propres. Ainsi, la première image de la base montre 65 % de la variance, la seconde 28 %, la troisième 6 % et la quatrième 1 %. Ces deux dernières quantités étant négligeables, les troisième et quatrième images peuvent être éliminées de la base et les images peuvent être reconstruites de façon satisfaisante à partir de la base ainsi réduite à ses deux premières images telles sur la figure 3.05 [16].

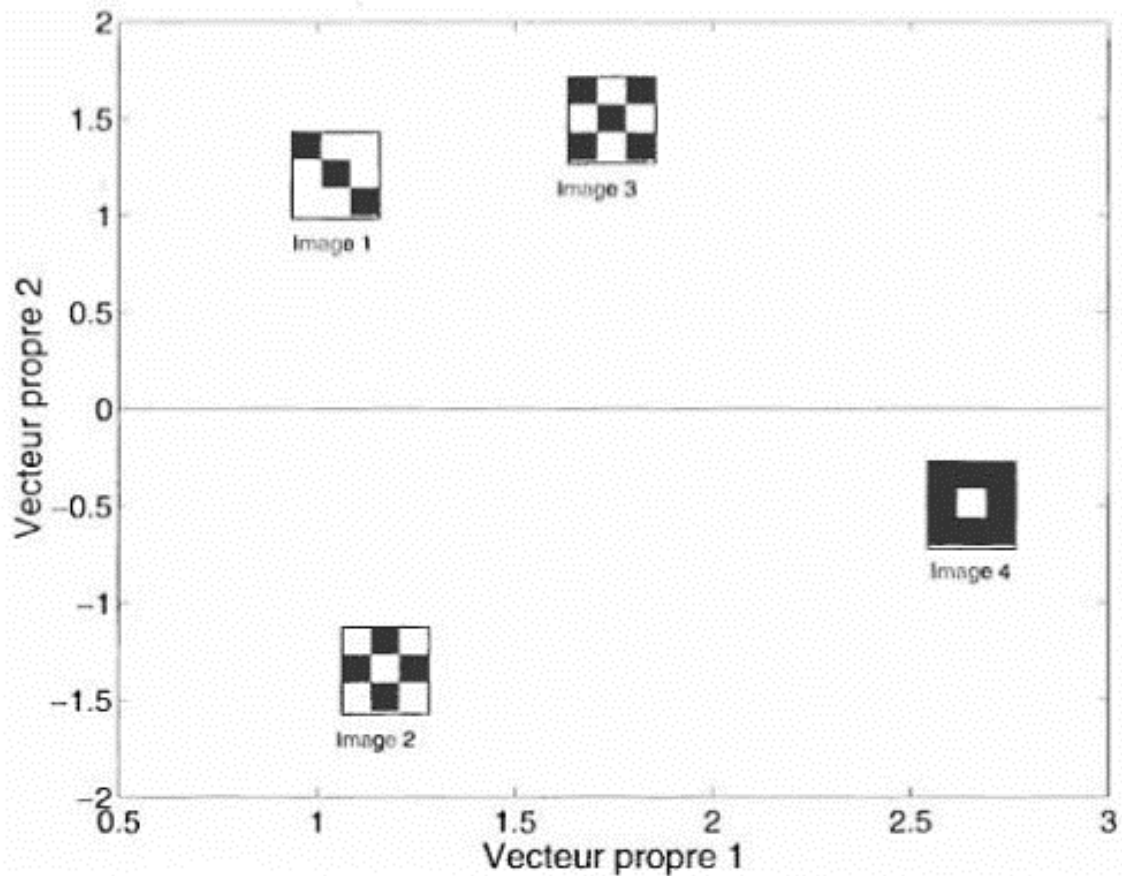


**Figure 3.04 :** *Reconstruction des images comme une somme pondérée des images de base.*



**Figure 3.05 :** *Estimation des images à partir des deux premières images de base.*

La projection des images sur les deux vecteurs propres peut être représentée graphiquement, comme illustrée par la figure 3.06. Cette figure permet d'établir une typologie des images. En effet, deux images présentant des caractéristiques communes sont proches l'une de l'autre. A l'opposé, deux images éloignées l'une de l'autre présentent des caractéristiques antagonistes. Par exemple, alors que toutes les images se projettent du même côté du premier vecteur propre, le deuxième vecteur propre oppose les images 1 et 3 aux images 2 et 4.



**Figure 3.06 :** *Projection des quatre images sur les deux premiers vecteurs propres de X.*

### 3.2.6 Image supplémentaire

Parfois, il arrive que l'on dispose d'images supplémentaires, mais que l'on ne veut pas, ou que l'on ne peut pas, prendre en compte ces images directement dans l'analyse. La technique des points supplémentaires permet de placer ces images dans l'espace défini par les premiers vecteurs propres. Cette technique consiste simplement à calculer les vecteurs propres de la matrice d'images et projeter les images supplémentaires (c'est-à-dire non prise en compte lors de l'analyse) sur les vecteurs propres précédemment calculés [18]. La technique des points supplémentaires permet,

entre autres, de vérifier si les vecteurs propres calculés à partir d'un échantillon d'images peuvent être utilisés comme base de reconstruction de nouvelles images.

### 3.3 ACP en reconnaissance faciale

Le but de l'approche ACP (ou des Visages Propres, eigenfaces en anglais) est de capturer la variation dans une collection d'images de visages afin d'utiliser cette information pour coder et comparer les visages. Etant donné un ensemble de différentes images de visages, la technique recherche les composantes principales de la distribution de visages puis l'exprime en termes de vecteurs propres. Chaque visage particulier est alors estimé par une approximation linéaire des vecteurs propres associés aux plus grandes valeurs propres.

#### 3.3.1 Principes et méthodes eigenfaces

L'idée principale consiste à exprimer un nombre  $N$  d'images d'apprentissage selon une base de vecteurs orthogonaux particuliers, contenant des informations indépendantes d'un vecteur à l'autre. Ces nouvelles données sont donc exprimées d'une manière plus appropriée à la reconnaissance du visage.

Nous voulons extraire l'information caractéristique d'une image de visage, pour l'encoder aussi efficacement que possible afin de la comparer à une base de données de modèles encodés de manière similaire. En termes mathématiques, cela revient à trouver les vecteurs propres de la matrice de covariance formée par les différentes images de notre base d'apprentissage.

Une image  $I_i(m, n)$  est traitée comme un vecteur  $\Gamma_i(m \times n, 1)$  dans un espace vectoriel de grande dimension ( $D = m \times n$ ), par concaténation des colonnes.

Après avoir rassemblé nos  $M$  images dans une unique matrice, nous obtenons une matrice d'images  $\Gamma$ , où chaque colonne représente une image  $\Gamma_i$ .

$$\Gamma = \begin{pmatrix} a_{1,1} & b_{1,1} & \cdots & z_{1,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & b_{n,1} & \cdots & z_{n,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,m} & b_{1,m} & \cdots & z_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,m} & b_{n,m} & \cdots & z_{n,m} \end{pmatrix} \quad (3.14)$$

On calcule ensuite l'image moyenne  $\Psi$  de toutes les images collectées.

$$\Psi = \frac{1}{N} \sum_{i=1}^N \Gamma_i \quad (3.15)$$

Cette image peut être vue comme le centre de gravité du jeu d'images et qui est représenté par la figure 3.07.



**Figure 3.07 : Image moyenne**

On ajuste ensuite les données par rapport à la moyenne.

L'image moyenne est alors soustraite de chaque image avec la formule suivante :

$$\Phi_i = \Gamma_i - \Psi, \quad i = 1 \dots N \quad (3.16)$$

On calcule ensuite la matrice de covariance du jeu de données. Cette matrice peut être vue comme une matrice de moments d'ordre 2 :

$$C = \sum_{i=1}^N \phi_i \phi_i^T = AA^T, \quad A = [\Phi_1 \Phi_2 \dots \Phi_N] \quad (3.17)$$

La prochaine étape consiste à calculer les vecteurs propres et les valeurs propres de cette matrice de covariance  $C$  de taille  $(D \times D)$ , c'est-à-dire de l'ordre de la résolution d'une image.

Le problème est que parfois, cela peut être difficile et très long.

En effet, si  $D > N$  (si la résolution est supérieure au nombre d'images), il y aura seulement  $N-1$  vecteurs propres qui contiendront de l'information (les vecteurs propres restants auront des valeurs propres associées nulles).

Par exemple, pour 50 images de résolution  $180 \times 200$ , nous pourrions résoudre une matrice  $L$  de  $50 \times 50$  au lieu d'une matrice de  $36000 \times 36000$  pour ensuite prendre les combinaisons linéaires appropriées des images  $\Phi_i$ . Le gain de temps de calcul serait considérable, nous passerions d'une complexité de l'ordre du nombre de pixels dans une image à celle de l'ordre du nombre d'images [2] [15].

Les étapes du processus qui nous permettent d'accélérer les calculs sont décrits ci-dessous : considérons les vecteurs propres  $e_i$  de  $C = AA^T$ , associés aux valeurs propres  $\lambda_i$ .

On a :

$$Ce_i = \lambda_i e_i \quad (3.18)$$

Les vecteurs propres  $v_i$  de  $L = A^T A$ , associés aux valeurs propres  $\mu_i$  sont tels que :

$$Lv_i = \mu_i v_i \quad (3.19)$$

Soit :

$$AA^T Av_i = A\mu_i v_i \quad (3.20)$$

Puisque  $C = AA^T$ , nous pouvons simplifier :

$$C(Av_i) = \mu_i (Av_i) \quad (3.21)$$

De (3.18) et (3.19), nous voyons que  $Av_i$  et  $\mu_i$  sont respectivement les vecteurs propres et les valeurs propres de  $C$  :

$$\begin{cases} e_i = Av_i \\ \lambda_i = \mu_i \end{cases} \quad (3.22)$$

Nous pouvons donc trouver les valeurs propres de cette énorme matrice  $C$  en trouvant les valeurs propres d'une matrice  $L$  beaucoup plus petite. Pour trouver les vecteurs propres de  $C$ , il suffit juste de multiplier les vecteurs propres de  $L$  par la matrice  $A$ .

Les vecteurs propres trouvés sont ensuite ordonnés selon leurs valeurs propres correspondantes, de manière décroissante. Plus une valeur propre est grande, plus la variance capturée par le vecteur propre est importante. Cela implique que la majeure partie des informations est contenue dans les premiers vecteurs propres [2] [15].

La méthode de reconnaissance faciale par eigenfaces se fait étape par étape, de l'acquisition des images de visage, leur représentation en vecteur, jusqu'à la représentation de tous les vecteurs propres. L'organigramme de la méthode eigenfaces est représenté dans la figure 3.08 (dans la page suivante). La partie mathématique de chaque traitement de l'organigramme est représentée à droite, ce sont les modèles mathématiques qu'on a déjà expliqués précédemment.

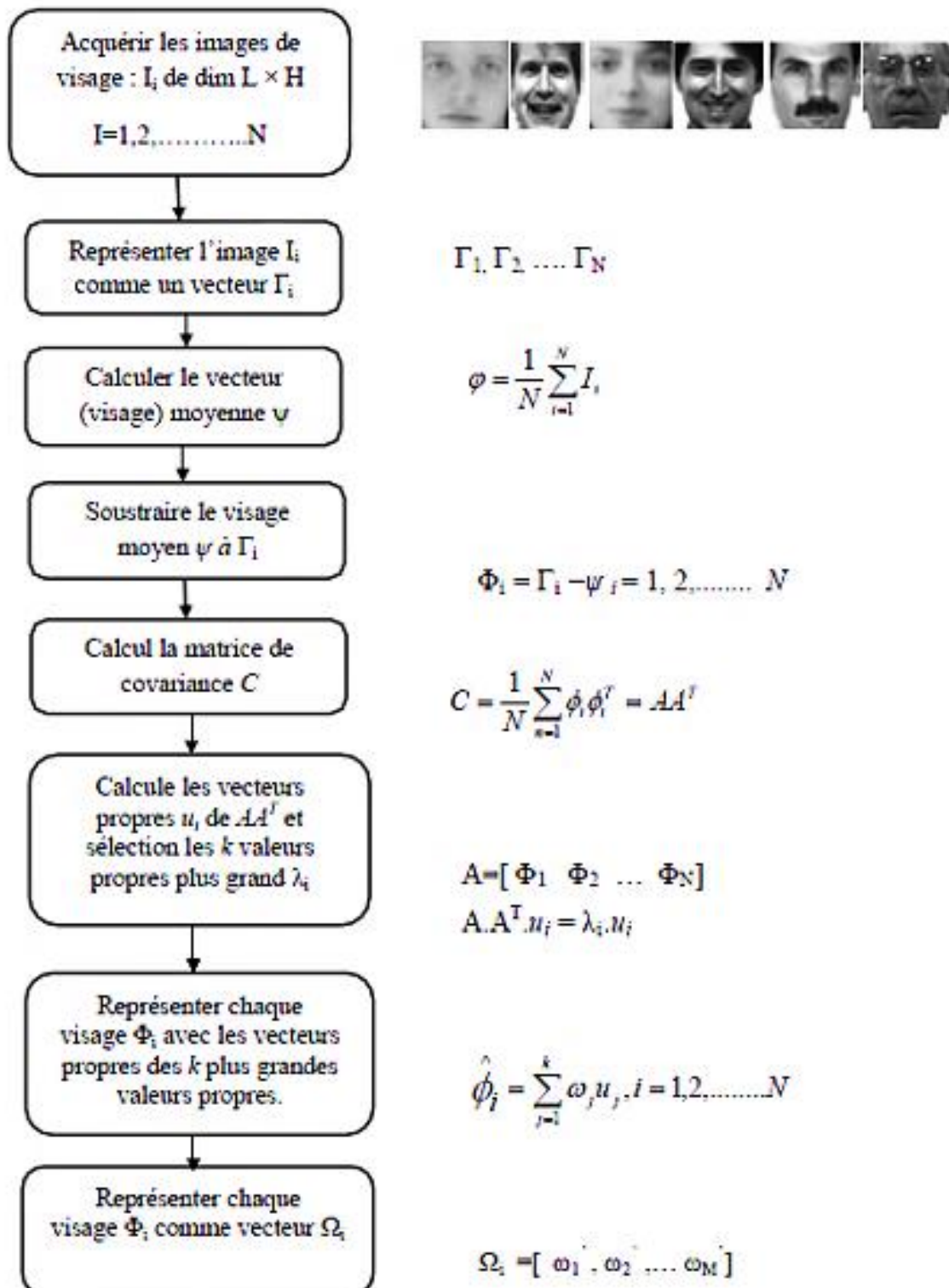


Figure 3.08 : Organigramme de la phase d'apprentissage d'ACP.

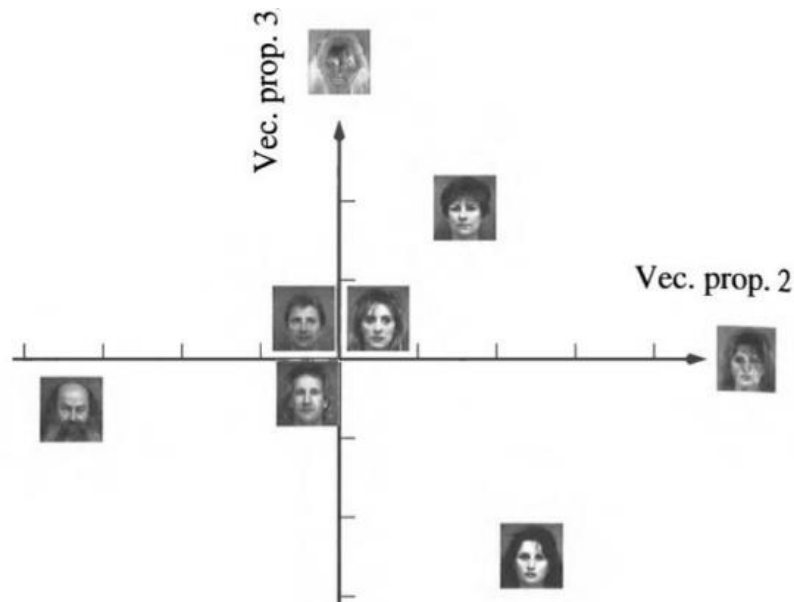
### 3.3.2 Projection des images de visage

De façon imagée, les vecteurs propres de l'ensemble de ses six visages présentés dans la figure 3.09 s'obtiennent comme une combinaison linéaire des six visages initiaux. Puis, on obtient six images par la soustraction des valeurs propres par ces visages initiaux et les prochains vecteurs propres s'obtiennent comme une combinaison linéaire de ses nouvelles images [16].



**Figure 3.09 :** *un ensemble de six images de visages*

Les techniques classiques d'interprétation d'ACP peuvent être utilisées pour interpréter les vecteurs propres d'un ensemble de visages. En particulier, il est possible de représenter les projections des visages sur des sous-plans à deux dimensions. Par exemple, la figure 3.10 montre la projection des six visages sur le sous-plan formé par les vecteurs propres 2 et 3. Il est clair, sur cette figure, que le second vecteur propre capture l'information liée au sexe des visages : il oppose les visages d'hommes aux visages de femmes. Le troisième vecteur propre semble opposer les cheveux longs aux cheveux courts.



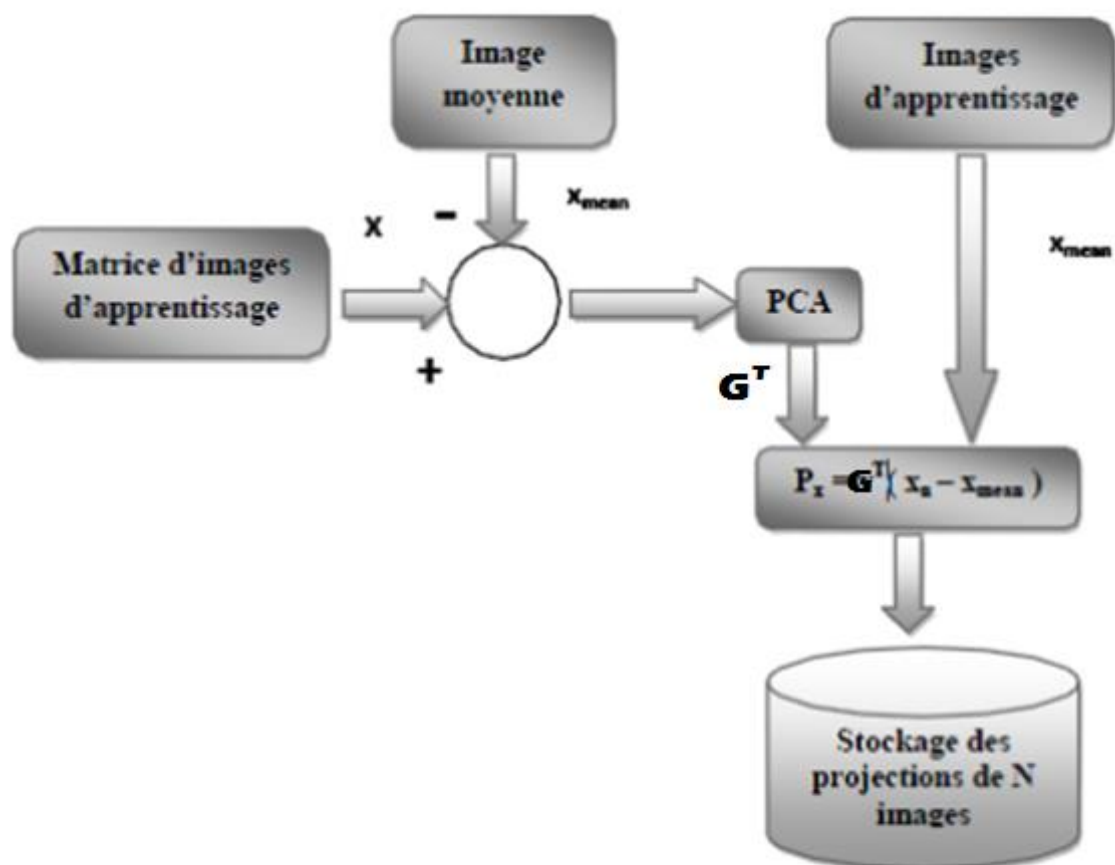
**Figure 3.10 :** *Projection des six visages sur le sous-plan formé par les vecteurs propres 2 et 3.*

### 3.3.3 Phase d'apprentissage

La phase d'apprentissage correspondrait à un enroulement réel de personnes qui seraient enregistrées dans une base de données. Cette phase consiste donc à récolter une grande quantité d'images de visage afin de se constituer une base de données de départ. Dans un premier temps, on construit une matrice contenant N images de la base d'apprentissage, puis l'image moyenne est calculée. On réajuste ensuite les données par rapport à la moyenne, pour pouvoir suivre, de manière simple, le comportement des valeurs d'écart type, de variance et de covariance. On applique alors un algorithme de reconnaissance globale à cette matrice réajustée.

La chose à retenir est que ces algorithmes fournissent en sortie ce que l'on appelle une matrice de projection G qui va nous être très utile dans la seconde partie de la phase d'apprentissage. Cette phase consiste en la projection des images apprises sur un espace vectoriel dont les vecteurs sont les éléments de notre matrice de projection G. Toutes ces projections sont finalement stockées dans une grande base de données [19].

La phase d'apprentissage (ou de modélisation) des eigenfaces se déroule comme sur la figure 3.11.



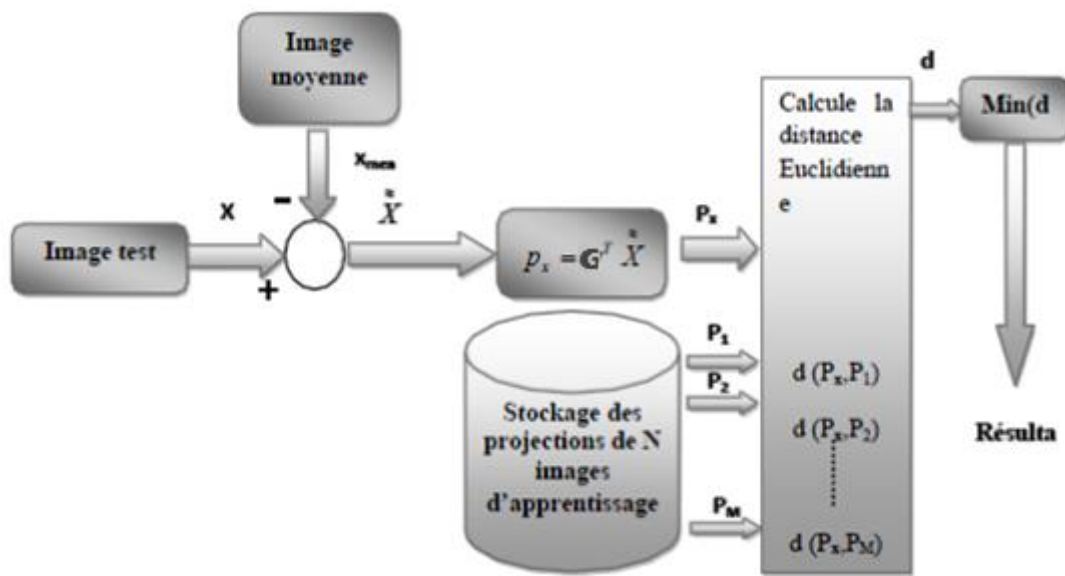
**Figure 3.11 : Phase d'apprentissage**



### 3.3.4 Phase de test

#### 3.4.4.1 Processus

Lorsqu'une nouvelle image de la base de test est mise devant le système, on la soustrait par rapport à la moyenne, on la projette ensuite sur l'espace vectoriel relatif à la matrice de projection  $G$  afin de la comparer avec toutes les projections issues de la phase d'apprentissage et qui étaient stockées dans la base de données. Par le terme comparer, il faut exécuter un calcul de distance entre les projections vectorielles. Il semble logique que plus la distance entre deux projections est petite, plus ces deux projections se ressemblent. Ainsi le résultat de la reconnaissance est l'image de la base d'apprentissage qui ressemble le plus à la nouvelle image présentée au système [19].

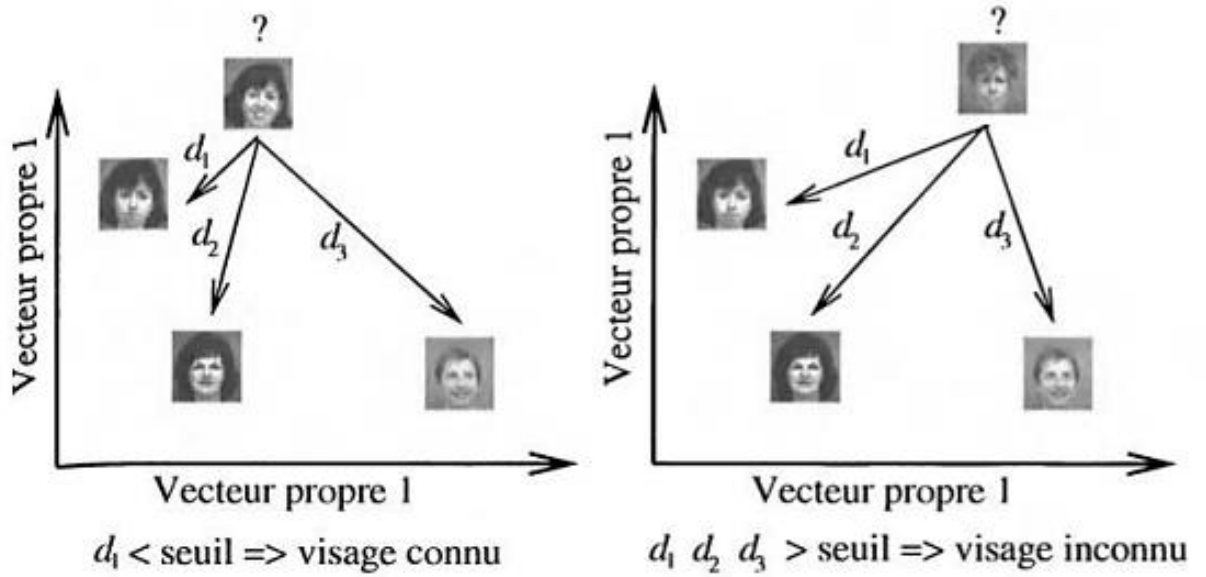


**Figure 3.12 : Phase de test (reconnaissance)**

#### 3.3.4.2 Mesure de distance

Lors de la phase de test, les visages appris, additionnés avec d'autres nouveaux visages qui sont généralement équivalents, sont reconstruits à partir des vecteurs propres déjà calculés. Un mécanisme de décision est ensuite mis en place pour décider quels visages parmi les visages reconstruits lors de la phase de test avaient été au préalable stockés en mémoire. L'algorithme le plus simple est l'algorithme du plus proche voisin. Il consiste à rechercher les visages les plus semblables aux visages tests sur l'hyperplan défini par les vecteurs propres (souvent, seuls les vecteurs propres ayant les plus grandes valeurs propres sont conservés). Ensuite, on calcule la distance euclidienne entre la projection du visage test et la projection de tous les visages appris. Si la distance entre le plus proche voisin (ou la distance moyenne entre les  $K$  plus proches voisins) et

le visage test est inférieure à un seuil donné, le visage est classé comme connu, sinon il est classé comme inconnu [16].



**Figure 3.13 :** Illustration de la simulation d'une tâche de reconnaissance à partir de l'algorithme du plus proche voisin

*a. Distance de Minkowski*

Lorsqu'on souhaite comparer deux vecteurs de caractéristiques issus du module d'extraction de caractéristiques d'un système biométrique, on peut soit effectuer une mesure de similarité (ressemblance), soit une mesure de distance (divergence).

La première catégorie de distances est constituée de distances Euclidiennes et sont définis à partir de la distance de Minkowski d'ordre  $p$  dans un espace euclidien (déterminant la dimension de l'espace euclidien) [2] [15].

Considérons deux vecteurs  $V = (v_1, v_2, \dots, v_n)$  et  $W = (w_1, w_2, \dots, w_n)$ , la distance de Minkowski d'ordre  $p$  noté  $L_p$  est défini par :

$$L_p = \left( \sum_{i=1}^n (|v_i - w_i|)^p \right)^{1/p} \quad (3.23)$$

Pour  $p = 2$ , on obtient la distance euclidienne :

$$L_2(v, w) = \sqrt{\sum_{i=1}^n |v_i - w_i|^2} \quad (3.24)$$

### *b. Distance dans l'espace de Mahalanobis*

En sortie de l'algorithme ACP, nous obtenons des vecteurs propres associés à des valeurs propres (représentant la variance selon chaque dimension). Ces vecteurs propres définissent une rotation vers un espace dont la covariance entre les différentes dimensions est nulle. L'espace de Mahalanobis est un espace où la variance selon chaque dimension est égale à 1. On l'obtient à partir de l'espace des images  $Im$  en divisant chaque vecteur propre par son écart-type correspondant.

Les vecteurs  $u$  et  $v$  sont reliés aux vecteurs  $r$  et  $s$  à partir des relations suivantes :

$$r_i = \frac{u_i}{\sigma_i} = \frac{u_i}{\sqrt{\lambda_i}} \text{ et } s_i = \frac{v_i}{\sigma_i} = \frac{v_i}{\sqrt{\lambda_i}} \quad (3.25)$$

Avec  $u$  et  $v$  deux vecteurs propres de l'image  $Im$ , issus de l'algorithme ACP,  $r$  et  $s$  deux vecteurs. Ainsi que  $\lambda_i$  les valeurs propres associées aux vecteurs  $u$  et  $v$ , et  $\sigma_i$  l'écart-type.

La distance Mahalanobis  $L2$  ( $Mah_{L2}$ ) est identique à la distance euclidienne à part qu'elle est calculée dans l'espace de Mahalanobis [2] [15]. Ainsi, pour des vecteurs propres  $u$  et  $v$  de projections respectives  $r$  et  $s$  sur l'espace de Mahalanobis, la distance Mahalanobis  $L2$  est définie par :

$$Mah_{L2}(u, v) = \sqrt{\sum_{i=1}^N |r_i - s_i|^2} \quad (3.26)$$

## **3.4 Conclusion**

Si ce chapitre montre le calcul de l'ACP par méthode matricielle, il donne aussi les modèles mathématiques correspondants. Etant un système d'analyse de donnée, l'ACP peut analyser les représentations matricielles des visages en utilisant leurs visages propres ; et prend ainsi le nom de méthode eigenfaces pour la reconnaissance faciale. La base d'apprentissage forme les eigenfaces de la base de visages et la reconnaissance se fait par calcul de la distance euclidienne entre ses bases et le visage test. Dans le dernier chapitre, on utilise Matlab pour faire la conception de notre système de reconnaissance faciale.

## CHAPITRE 4

### SIMULATION RECONNAISSANCE FACIALE HYBRIDE

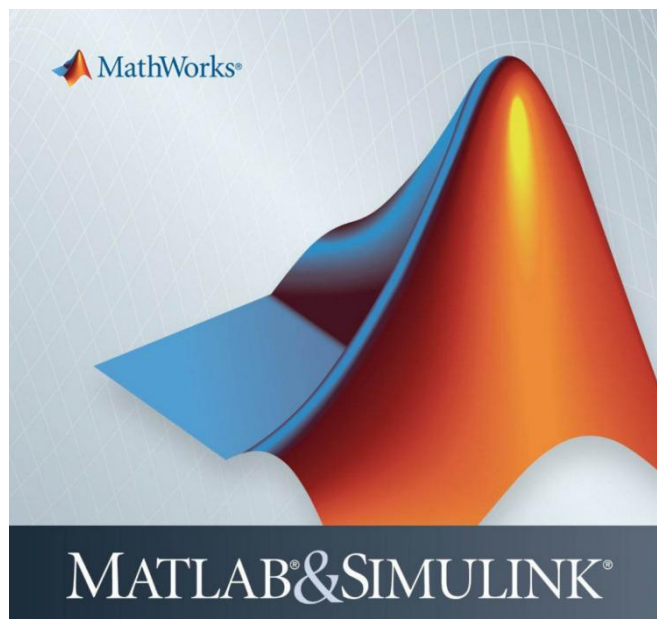
#### 4.1 Présentation du logiciel MATLAB

##### 4.1.1 Introduction

MATLAB est un système interactif et convivial de calcul numérique et de visualisation graphique destiné aux ingénieurs et scientifiques. Il possède un langage de programmation à la fois puissant et simple d'utilisation. Il permet d'exprimer les problèmes et solutions d'une façon aisée, contrairement aux autres langages de programmation.

Il s'impose dans le monde universitaire et industriel comme un outil puissant de simulation et de visualisation de problèmes numériques. Dans le monde universitaire, MATLAB est utilisé pour l'enseignement de l'algèbre linéaire, le traitement du signal, l'automatique, ainsi que dans la recherche scientifique. Dans le domaine industriel, il est utilisé pour la résolution et la simulation de problèmes pratiques d'ingénierie et de prototypage.

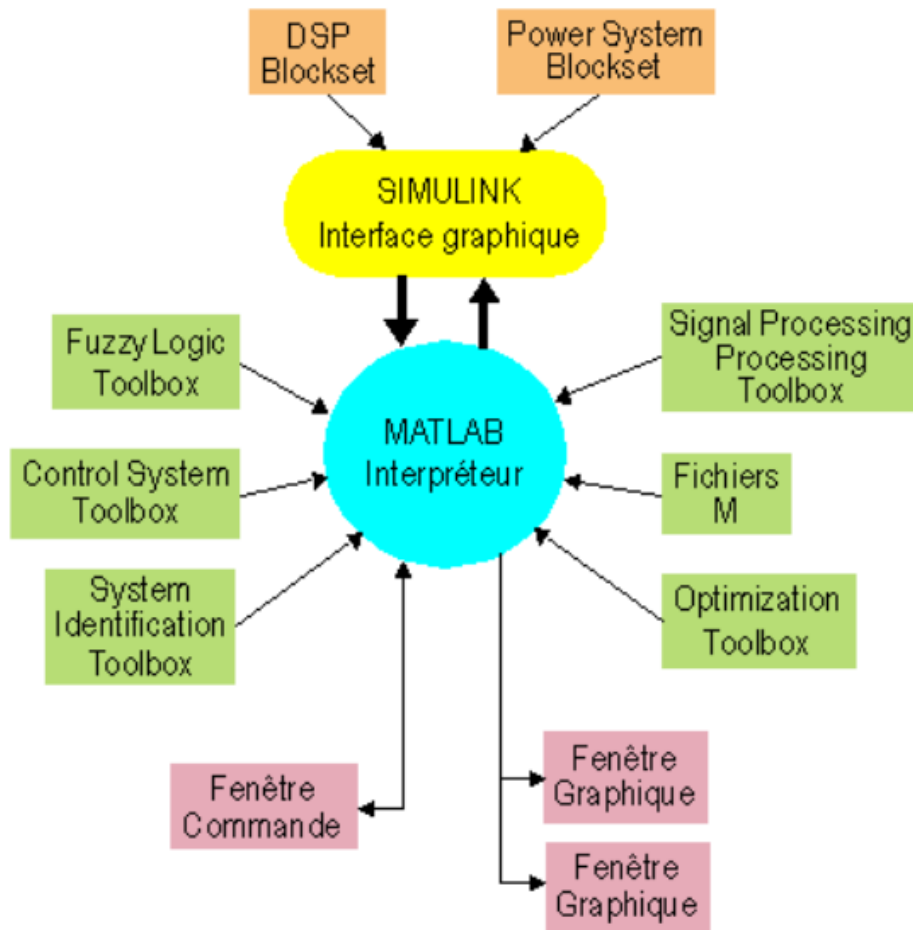
Ecrit à l'origine, en fortran, par Cleve Moler, MATLAB était destiné à faciliter l'accès au logiciel matriciel développé dans les projets LINPACK et EISPACK. La version actuelle, écrite en C par The MathWorks Inc., existe en version « professionnelle » et en version « étudiant ». Sa disponibilité est assurée sur plusieurs plates-formes : Sun, Bull, HP, IBM, compatibles PC, Macintosh, et plusieurs machines parallèles [20].



**Figure 4.01 :** *Logo Matlab*

#### 4.1.2 Environnement Matlab

Matlab fonctionne dans plusieurs styles d'environnement, on peut voir ces différents environnements dans la figure 4.02 [21].



**Figure 4.02 : Environnement MATLAB**

**Fenêtre Commande :** Dans cette fenêtre, l'utilisateur donne les instructions et MATLAB retourne les résultats.

**Fenêtres Graphique :** MATLAB trace les graphiques dans ces fenêtres.

**Fichiers M :** Ce sont des programmes en langage MATLAB (écrits par l'utilisateur).

**Toolboxes :** Ce sont des collections de fichiers M développés pour des domaines d'application spécifiques (Signal Processing Toolbox, System Identification Toolbox, Control System Toolbox,  $\mu$ -Synthesis and Analysis Toolbox, Robust Control Toolbox, Optimization Toolbox, Neural Network Toolbox, Spline Toolbox, Chemometrics Toolbox, Fuzzy Logic Toolbox, etc.)

**Simulink :** C'est l'extension graphique de MATLAB permettant de travailler avec des diagrammes en blocs.

Blocksets : Ce sont des collections de blocs Simulink développés pour des domaines d'application spécifiques (DSP Blockset, Power System Blockset, etc.).

## **4.2 Commande de base dans Matlab**

### **4.2.1 Aide dans MATLAB**

Le premier écran de MATLAB présente quelques commandes : intro, demo, help help, help, info.

intro : lance une introduction à MATLAB,

help : produit une liste des fonctions MATLAB par catégorie,

help help : informations sur l'utilisation de l'aide,

info : informations sur les boîtes à outils (Toolboxes) disponibles,

Matlab possède un très grand nombre de fonctions dans chaque toolboxes, ceci dépend seulement de ce que l'on veut faire : Traitement de signal, d'image, réseau de neurones, etc.

Mais dans ce chapitre, on ne présentera que quelques fonctions qui nous seront utiles.

### **4.2.2 Fonctions sur les vecteurs et matrices**

Dans MATLAB, il y a un seul type de données : le type matrice (Matrix). Tout est matricé, un scalaire est une matrice carrée d'ordre 1. Il n'y a donc pas de déclaration de types. De même, l'utilisateur ne s'occupe pas de l'allocation mémoire. Les variables matrices et vecteurs peuvent être redimensionnées et même changer de type [20].

MATLAB dispose de fonctions qui opèrent directement sur les vecteurs ou les colonnes d'une matrice. Parmi celles-ci, on peut citer :

mean : valeur moyenne,

std : écart type,

sum : somme,

cumsum : somme cumulée,

cumprod : produit cumulé,

min : valeur minimale,

max : valeur maximale,

diff : différence des éléments successifs.

prod : produit,

sort : ordre croissant ou décroissant des éléments du vecteur.

## 4.3 Détections du visage et les parties du visage

### 4.3.1 Principe de l'algorithme Viola Jones

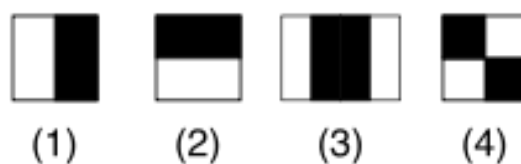
Une avancée majeure dans le domaine de la détection de visage a été réalisée par Viola et Jones en 2001. Ces derniers ont proposé une méthode basée sur l'apparence (« Appearance-based methods »), robuste et tournante à 15 fps pour des images de 384 x 288 pixels sur un pc Intel Pentium III 700 Mhz. Ce fut la première méthode en temps réel présenté. La renommée de cette approche est faite sur trois concepts : l'image intégrale, l'algorithme d'apprentissage basé sur Adaboost et la détection cascade [22] [23].

#### 4.3.1.1 Image intégrale

Cette méthode permet d'analyser une image, sans avoir besoin d'étudier chaque pixel la constituant. En effet, la notion d'image intégrale permet de définir plusieurs zones rectangulaires au sein d'une image. L'intérêt de cette technique réside dans le fait qu'elle offre la possibilité d'accéder à la valeur des autres zones à gauche et au-dessus de la zone sur laquelle nous sommes.

L'algorithme se base sur les caractéristiques de Haar (Haar features) pour localiser les visages présents sur une image d'entrée. Dans le but d'extraire rapidement ces caractéristiques, l'image est représentée sous forme intégrale. En effet, sous cette forme, l'extraction d'une caractéristique à n'importe quel endroit et à n'importe quelle échelle est effectuée en un temps constant tandis que le temps de conversion vers la représentation intégrale ne remet pas en cause ce gain de temps offert par l'utilisation de la représentation en image intégrale. La définition des caractéristiques de Haar et la manière dont la représentation intégrale accélère considérablement leur extraction sont présentées ci-après pour une image en niveaux de gris.

Dans toute image, une zone rectangulaire peut être délimitée et la somme des valeurs de ses pixels calculée. Une caractéristique de Haar est une simple combinaison linéaire de sommes ainsi obtenues. Plusieurs caractéristiques de Haar peuvent être définies selon le nombre, les échelles, les positions et les dimensions des zones rectangulaires considérées. On a ici 4 exemples de caractéristiques de Haar présentés à la figure 4.03.



**Figure 4.03 :** Exemple de 4 caractéristiques de Haar (utilisée par Viola Jones)

La somme des valeurs des pixels appartenant aux zones encadrées claires est soustraite à la somme des valeurs des pixels appartenant aux zones encadrées sombres pour obtenir la caractéristique de Haar. Chacune des quatre caractéristiques de Haar est représentée avec son cadre de détection respectif.

Présenté comme tel, le calcul d'une caractéristique de Haar demande à chaque fois l'accès aux valeurs de tous les pixels contenus dans les zones rectangulaires considérées.

Cela devient vite contraignant temporellement dès que les caractéristiques de Haar sont définies par des zones rectangulaires de grandes dimensions. L'image intégrale permet de surmonter ce problème en rendant constant le temps de calcul d'une caractéristique de Haar à n'importe quelle échelle. L'image intégrale est représentée mathématiquement par :

$$i(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (4.01)$$

$$\forall 0 < x \leq width, 0 < y \leq height \quad (4.02)$$

Avec  $i(x, y)$  est l'image d'origine et  $i(x', y')$  l'image sous sa nouvelle représentation. Ainsi chaque pixel a pour valeur la somme des valeurs des pixels compris dans le rectangle défini par le coin supérieur gauche de l'image et lui-même.

Le calcul de la somme des valeurs des pixels appartenant à une zone rectangulaire s'effectue en accédant seulement à quatre pixels de l'image intégrale : soit un rectangle ABCD dont les sommets sont nommés dans le sens des aiguilles d'une montre en commençant par le sommet supérieur gauche et soit a, b, c, d les valeurs sous la représentation intégrale respective de A, B, C, D. La somme des valeurs des pixels appartenant à ABCD est donnée par  $c - b - d + a$ , quelle que soit sa taille.

Une caractéristique de Haar étant une combinaison linéaire de tels rectangles ABCD, son calcul se fait alors en un temps indépendant de sa taille [23].

#### 4.3.1.2 Algorithme d'apprentissage basé sur Adaboost

Pour localiser les visages sur l'image d'entrée, cette dernière est scannée par une fenêtre de dimension déterminée. La fenêtre parcourt l'image et son contenu est analysé pour savoir s'il s'agit d'un visage ou non. Comme dit plus haut, les caractéristiques de Haar sont extraites pour effectuer la classification et de ce fait la représentation intégrale de l'image accélère l'analyse. Mais, pour une fenêtre de 24x24 pixels il y a 45.396 caractéristiques de Haar, les traiter toutes prendrait beaucoup trop de temps pour une application en temps réel. Pour surmonter ce problème, une variante de la méthode de boosting Adaboost est utilisée.

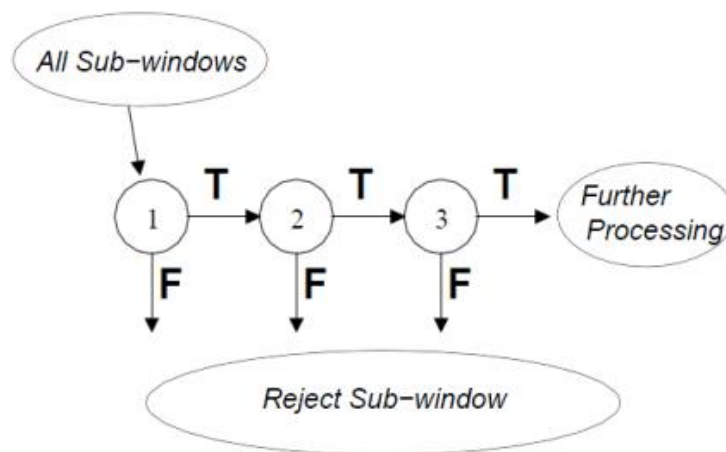


L'algorithme Adaboost constitue le deuxième apport du travail de Viola & Jones.

Adaboost est une méthode d'apprentissage permettant de « booster » les performances d'un classificateur quelconque nommé « classificateur faible ». L'idée est de faire passer les candidats à classifier à travers plusieurs classificateurs faibles, chacun étant entraîné en portant plus d'attention sur les candidats mal classifiés par le classificateur précédent [23].

#### 4.3.1.3 Détection Cascade

L'idée de base derrière le concept de Cascade est que parmi l'ensemble des candidats, c'est à dire l'ensemble des états de la fenêtre de recherche, une partie peut être éliminée sur base de l'évaluation de seulement quelques caractéristiques de Haar. Une fois cette élimination effectuée, les candidats restants sont analysés par des classificateurs forts plus complexes (utilisant plus de caractéristiques de Haar) demandant un plus grand temps de traitement. En utilisant plusieurs étages de ce type, le processeur évite d'effectuer des analyses lourdes en temps de calcul sur des échantillons pour lesquels il est rapidement possible de se rendre compte qu'ils sont négatifs. Le processus de classification apparaît alors comme une cascade de classificateurs forts de plus en plus complexes où à chaque étage les échantillons classifiés négatifs sont sortis tandis que les échantillons classifiés positifs sont envoyés aux classificateurs suivants [23]. Ceci est représenté à la figure 4.04. A chaque étage, seuls les candidats classifiés positifs sont transmis à l'étage suivant.



**Figure 4.04 :** *Cascade de classificateurs forts.*

Si le premier étage rejette un faux négatif, c'est un gros problème, car il ne sera jamais récupéré par la cascade. Autrement dit c'est un visage qui ne sera pas détecté. Par contre, si le premier étage transmet un faux positif, il pourra toujours être éliminé aux étages suivants de la cascade. Ce petit raisonnement permet de mettre en évidence que les premiers nœuds constitutifs de la cascade

peuvent se permettre d'avoir un taux de faux positifs élevés (de l'ordre de 40-50%), mais doivent absolument assurer un taux de détection maximum.

Ce concept permet donc à l'algorithme de consacrer son temps à de longues analyses complexes uniquement lorsque cela en vaut la peine. Il s'agit donc à nouveau d'un mécanisme qui accélère la vitesse d'exécution de la méthode proposée par Viola & Jones.

#### **4.3.2 Détection de visage avec la fonction : *CascadeObjectDetector***

##### **4.3.2.1 Description**

Cette méthode appartient au toolbox « Computer Vision System » de Matlab, dans le package « vision ». Le détecteur d'objet en cascade utilise l'algorithme Viola-Jones pour détecter les visages des gens, le nez, les yeux, la bouche ou le haut du corps. On peut également utiliser la fonction *trainCascadeObjectDetector* pour former un classificateur personnalisé à utiliser avec ce système [26].

##### **4.3.2.2 Construction**

Voici les syntaxes pour utiliser la fonction *vision.CascadeObjectDetector* [26] :

- *detector = vision.CascadeObjectDetector* est la méthode utilisée pour détecter les objets avec l'algorithme Viola-Jones. La propriété du *ClassificationModel* contient le type d'objet à détecter. Par défaut, le détecteur est configuré pour détecter les visages.
- *detector = vision.CascadeObjectDetector(MODEL)*, si on veut spécifier l'objet à détecter, on le définit par la variable d'entrée *MODEL*. Il y a plusieurs type de *MODEL* valides, telles que '*FrontalFaceCART*' pour le visage vu de face, '*UpperBody*' pour la partie en haut du corps, '*ProfileFace*' pour le visage en profil, etc.
- *detector = vision.CascadeObjectDetector(XMLFILE)*, dans cette syntaxe, la configuration utilise le modèle de classification personnalisé spécifié avec le fichier *XMLFILE*. Le *XMLFILE* peut être créé en utilisant la fonction *trainCascadeObjectDetector* ou la fonction d'apprentissage d'OpenCV (Open Source Computer Vision). Il faut spécifier un chemin complet ou relatif du *XMLFILE*.
- *detector = vision.CascadeObjectDetector (Name, Value)*, cette syntaxe est utilisée pour configurer les propriétés de la détection de l'objet en cascade. On doit spécifier ces propriétés comme un ou plusieurs arguments de paires « nom-valeur ».

#### 4.3.2.3 Détection de l'objet

Premièrement, il faut définir et mettre en place le détecteur d'objet en cascade à l'aide du constructeur. Ensuite, on appelle la fonction *step* avec l'image d'entrée I, l'objet de détection en cascade *detector*, les points spécifiques, et toutes les propriétés facultatives.

Voici les syntaxes pour utiliser la fonction *step* [26] :

- $BBOX = step(detector, I)$  renvoie *BBOX*, une matrice  $M \times 4$  définissant M zone rectangulaire contenant les objets détectés. Cette méthode effectue la détection d'objet multiscale sur l'image entrée I. Chaque ligne de la matrice de sortie *BBOX*, contient un vecteur de quatre éléments [x y largeur hauteur] en pixels, qui spécifie le coin supérieur gauche (de coordonnées x et y) et la taille de la zone de sélection (largeur et hauteur). L'image entrée I, doit être une image à niveau de gris ou une image true color (RGB).
- $BBOX = step(detector, I, Z)$  est utilisé si on veut détecter les objets dans une zone de recherche spécifiée définie par Z. Il faut préciser Z comme un vecteur à 4 éléments [x y largeur hauteur], qui déterminent une zone rectangulaire à l'intérêt de l'image I.

### 4.4 ACP sous MATLAB

#### 4.4.1 Fonction Eig

##### 4.4.1.1 Syntaxe

Voici les syntaxes possibles pour utiliser la fonction *eig* de Matlab [26] :

$$e = eig(A)$$

$$[V, D] = eig(A)$$

$$[V, D, W] = eig(A)$$

$$e = eig(A, B)$$

$$[V, D] = eig(A, B)$$

$$[V, D, W] = eig(A, B)$$

$$[ \_ ] = eig(A, balanceOption)$$

$$[ \_ ] = eig( \_, eigvalOption)$$

##### 4.4.1.2 Description

$e = eig(A)$  renvoie les valeurs propres de la matrice carrée A sous forme d'un vecteur colonne e.

$[V, D] = eig(A)$  renvoie une matrice diagonale D et une matrice complète V qui représentent respectivement les valeurs propres et les vecteurs propres de A, de sorte que  $A * V = V * D$ .

$[V, D, W] = eig(A)$  renvoie  $V$ ,  $D$  et également la matrice complète  $W$ , les colonnes de  $W$  sont les vecteurs propres correspondant à gauche, de sorte que  $W^*A = D^*W$ .

$e = eig(A, B)$  renvoie les valeurs propres généralisées des matrices carrées  $A$  et  $B$  dans un vecteur colonne  $e$ .

$[V, D] = eig(A, B)$  renvoie la matrice diagonale  $D$  et la matrice complète  $V$  qui contiennent respectivement les valeurs propres généralisées et les vecteurs propres correspondant à droite des deux matrices carrées  $A$  et  $B$ , en suivant la propriété  $A^*V = B^*V^*D$ .

$[V, D, W] = eig(A, B)$ , en plus de  $V$  et  $D$ , cette syntaxe renvoie également la matrice complète  $W$ , les colonnes de  $W$  sont les vecteurs propres correspondant à gauche, de sorte que  $W^*A = D^*W^*B$ .

$[ \_ ] = eig(A, balanceOption)$ , si l'argument `balanceOption` est 'nobalance', ceci désactive l'étape d'équilibrage préliminaire dans l'algorithme. Sinon, la valeur par défaut pour `balanceOption` est 'balance', qui permet l'équilibrage. La fonction *eig* peut renvoyer l'un des arguments de sortie dans syntaxes précédentes.

$[ \_ ] = eig(\_, eigvalOption)$ , on ajout l'entrée `eigvalOption` si on veut spécifié la forme des valeurs propres, et en utilisant l'un des arguments d'entrée et de sortie dans les syntaxes précédentes. On définit `eigvalOption` par 'vector' pour renvoyer les valeurs propres dans un vecteur de colonne ou en 'matrix' pour retourner les valeurs propres dans une matrice diagonale [26].

#### **4.4.2 Distance euclidienne**

En suivant la formule de la distance euclidienne qu'on a vue dans le chapitre 3 sur les mesures de distance. Matlab peut calculer celui-ci avec la fonction `norm()`.

$n = norm(v)$  retourne la norme 2 ou plus exactement la distance euclidienne du vecteur  $v$  [26].

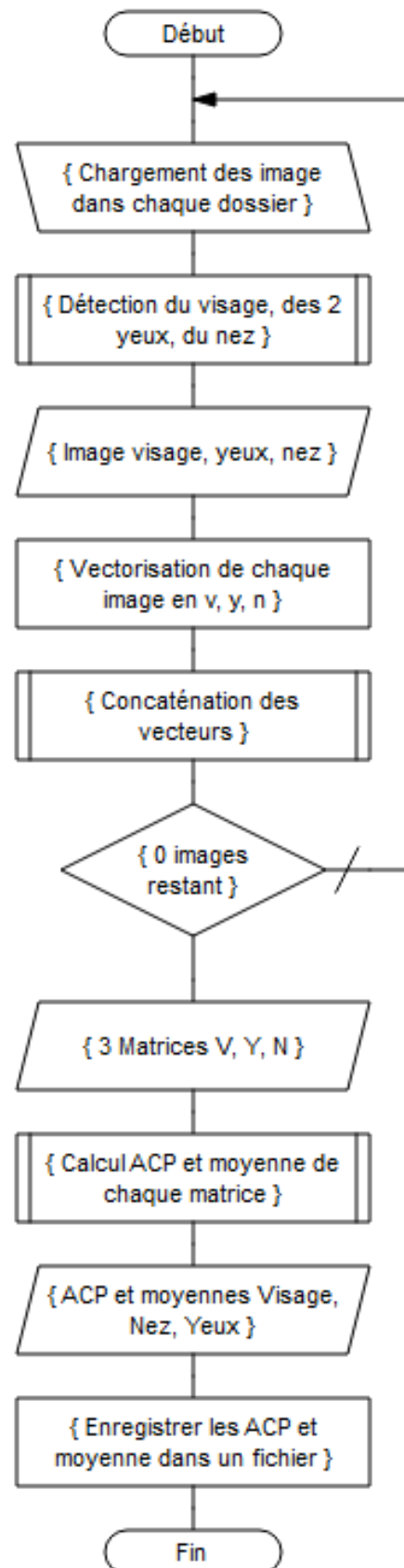
### **4.5 Reconnaissance de visage, des yeux et du nez**

#### **4.5.1 Organigramme du programme**

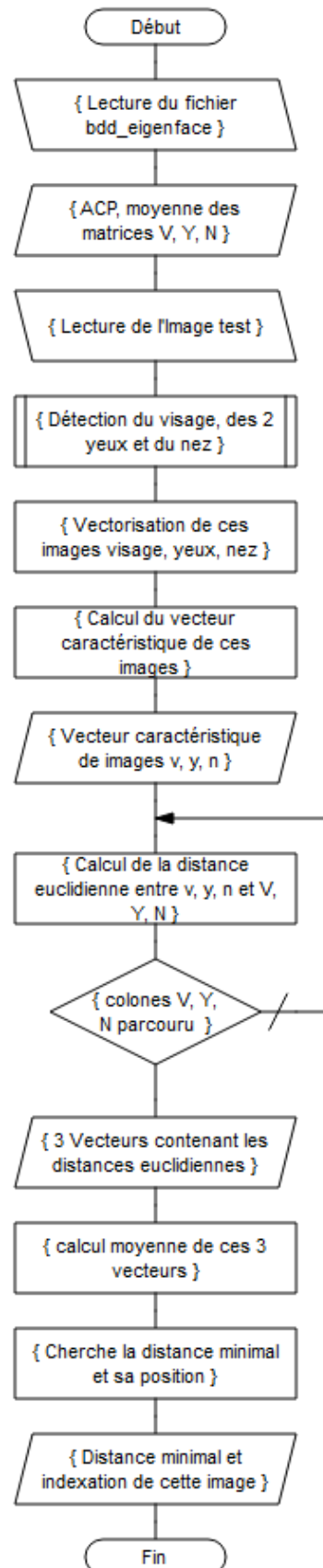
Les organigrammes qu'on présente ici sont les organigrammes principaux de notre projet. Ces organigrammes ont été tracés par « Monia Org 0.38 », c'est un logiciel d'initiation d'algorithme par organigramme.

On dispose ici de deux organigrammes principaux : dans la figure 4.05, on a l'algorithme d'apprentissage et de la création de la base d'eigenfaces du visage, des deux yeux et du nez. Dans la figure 4.06, on y voit l'algorithme de reconnaissance du visage, des deux yeux et du nez.

Et c'est avec la moyenne des ressemblances qu'on indexe le visage correspondant.



**Figure 4.05 :** Organigramme de création et apprentissage de la base d'eigenfaces



**Figure 4.06 :** *Organigramme de la reconnaissance*

#### **4.5.2 Base de visage standard: Collection d'images du visage Faces94**

##### **4.5.2.1 Description de la base Face94**

Il existe de nombreuses bases de visage qu'on utilise pour évaluer un système de reconnaissance faciale. Dans notre cas, on va utiliser une base de visage de Computer Vision Science Research Projects (Dr Libor Spacek) qu'on a téléchargée dans [27]. C'est une collection d'images du visage nommée Faces94, contenant 153 nombres de personnes avec des résolutions d'image : 180 par 200 pixels (format portrait) et ceci dispose de trois répertoires : femelle (20), male (113), malestaff (20). On trouve dans ses trois dossiers des sous-dossiers représentant chacun une personne, chaque personne dispose de 20 images dans les sous-dossiers.

Dans cet ensemble, on va réduire cette base de 100 personnes après avoir testé l'efficacité de la détection de visage « Viola Jones » sur les images et on a aussi réduit à 10 images pour chaque personne au maximal. On utilisera ces 10 images de chaque personne pour l'apprentissage et on prend le reste pour faire des tests. Il se peut que certaines personnes ne possèdent qu'une, deux ou trois images seulement dans leurs répertoires, c'est pour faire au mieux les évaluations.

##### **4.5.2.2 Système d'acquisition des visages**

Les sujets sont assis à distance fixe de l'appareil et sont invités à prendre la parole, tandis qu'une séquence d'images est prise. Le discours est utilisé pour introduire la variation d'expression du visage.

##### **4.5.2.3 Variation de la propriété de chaque image**

L'arrière-plan de toutes les images est vert. Il y a une variation très mineure au niveau de l'inclinaison et de la pente du visage. La position du visage dans l'image est peu modifiée.

Il y a des changements considérables d'expression sur les images de chaque individu.

Il n'y a pas de variation de cheveux des individus vu que les images ont été prises en une seule séance.

On peut voir dans cette figure 4.07 des images de visage de la personne du sous-dossier « cj carr ».



**Figure 4.07 : Exemple d'image de visage d'une personne**

### 4.5.3 Base de visage personnalisée

Pour évaluer la performance d'un système de reconnaissance faciale, il est recommandé d'utiliser des bases de visage standard comme on vient de présenter. Mais pour tester la capacité de cet algorithme, on utilisera aussi une base de données réelle, c'est-à-dire qu'on va créer une base de visage avec des images acquises personnellement.

Un sous-programme développé sur Matlab, utilise la webcam de l'ordinateur pour l'acquisition des images. La détection du visage se fait en temps réel, ainsi quand l'application aperçoit qu'on peut faire à la fois la détection du visage, des yeux et du nez, la capture se fait automatiquement.



**Figure 4.08 :** *Exemples d'image de visage capturé*

On montre dans cette figure 4.08 un exemple de visage qu'on a capturé avec le sous-programme. Les images de la base de visage personnalisé sont soumises à une légère variation de la lumière, la plupart avec des expressions différentes et certaines photos sont prises à des dates différentes. Ce sont des images couleur RGB et enregistré sous l'extension .jpg, de dimension 302x302 pixels.

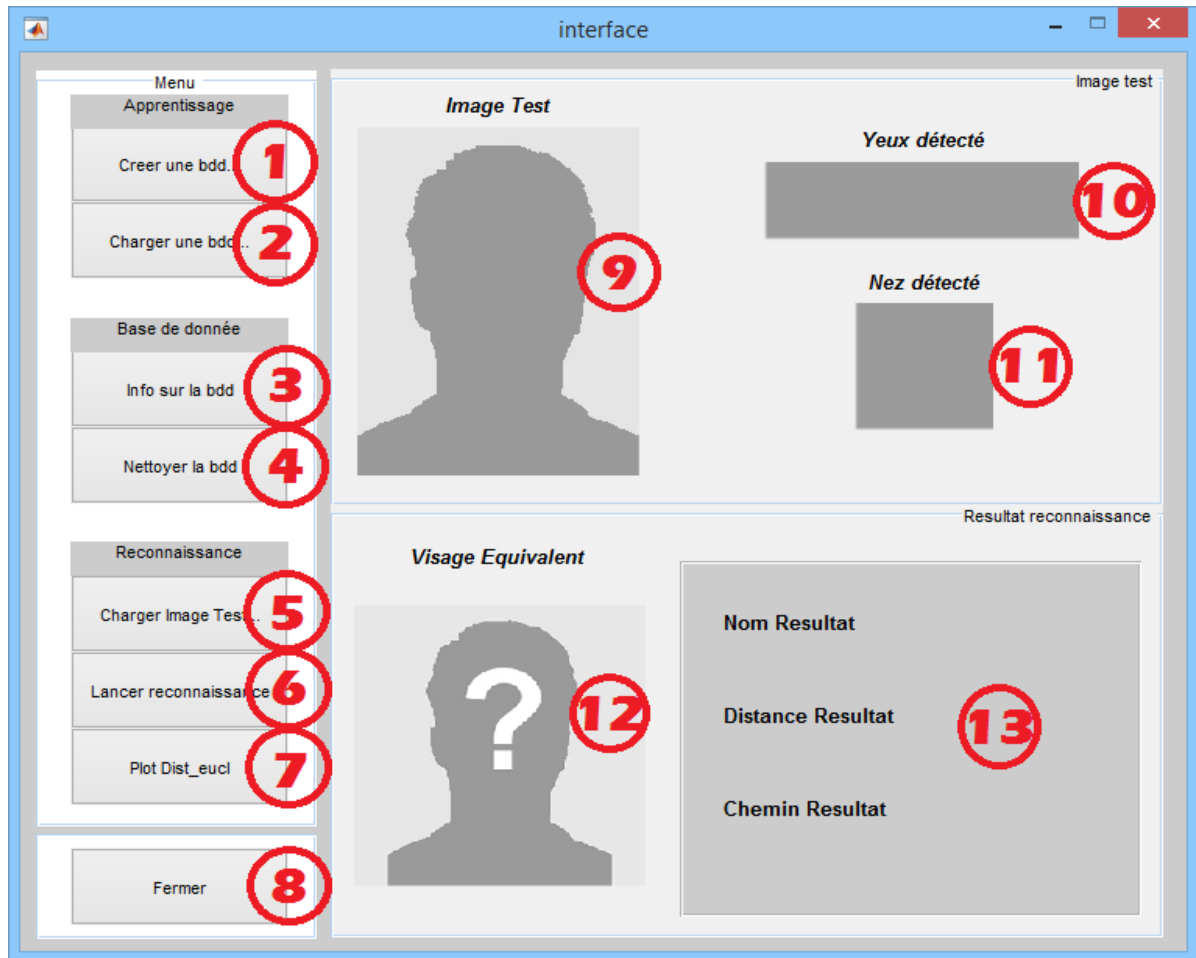
Les images sont déjà organisées selon notre programme d'apprentissage (numéroté de 1.jpg jusqu'au nombre d'images dans le dossier). Comme le système de base de données précédent, chaque personne est représentée par un dossier qui contient les images de visage. Et le nom du dossier indique le nom de la personne.

Il est possible de mettre autant de personnes dans la base de données. Mais il faut aussi prendre en compte que plus la base de visage sera lourde plus le système sera lent en apprentissage et en reconnaissance. Le nombre de personnes dans la base n'affecte pas vraiment la stabilité du système, cependant la performance de la reconnaissance dépend du nombre d'images de visage appris dans chaque dossier d'une personne.



#### 4.5.4 Interface de notre système de reconnaissance

On peut voir dans cette figure 4.09 l'interface et le menu principal de notre application.

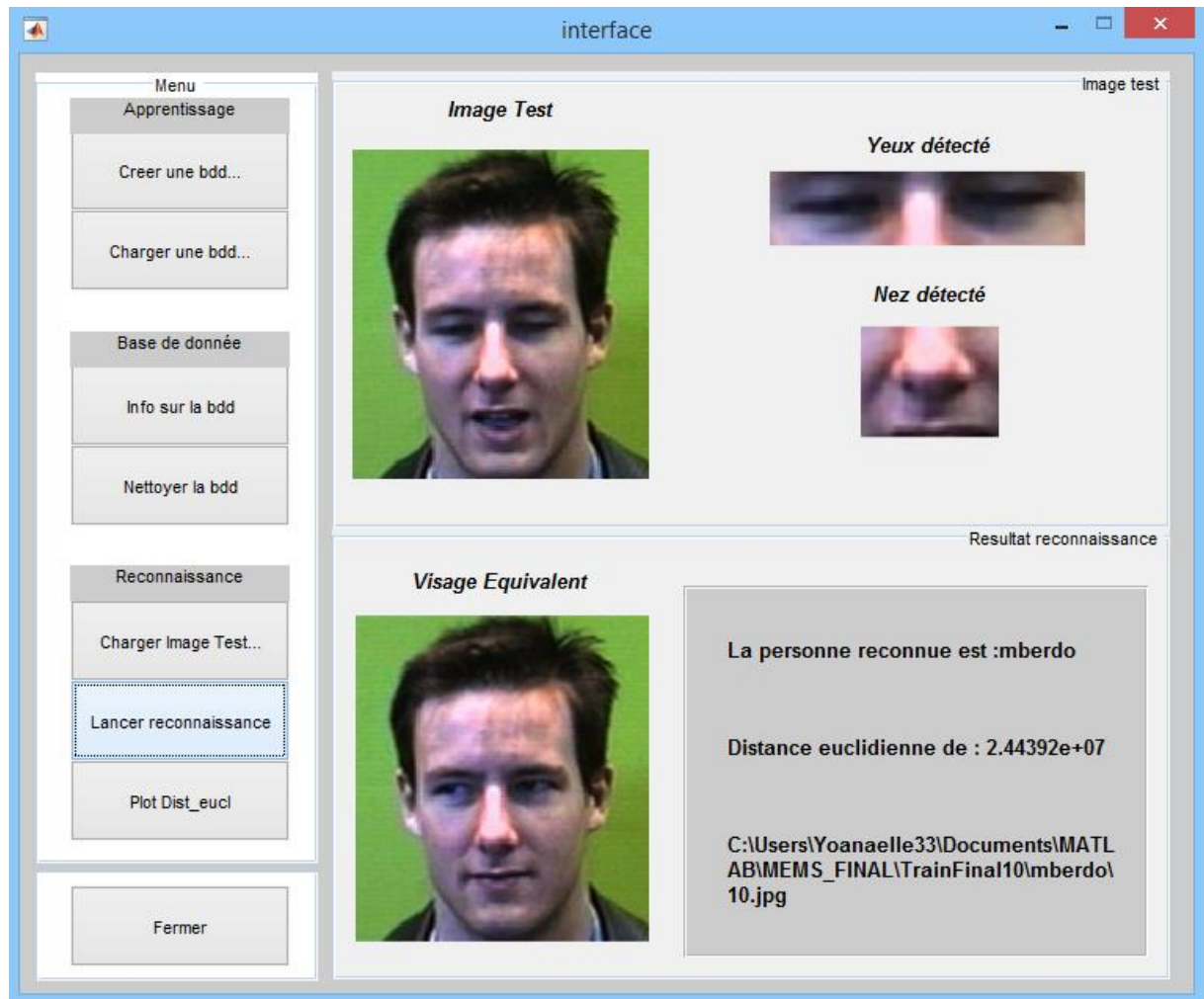


**Figure 4.09 : Interface et menu principal**

- 1) On charge un dossier pour créer une base de données
- 2) On charge une base de données déjà créée
- 3) Information sur la base de données chargée
- 4) Nettoyer une base de données
- 5) On charge un fichier image (.jpg) pour la reconnaissance
- 6) Lancer la reconnaissance
- 7) Afficher le graphe contenant toutes les distances euclidiennes
- 8) Quitter l'application
- 9) Aperçu de l'image test
- 10) Aperçu des yeux détectés
- 11) Aperçu du nez détecté

- 12) Aperçu du visage trouvé (équivalent)
- 13) Information sur le visage trouvé (équivalent)

#### 4.5.5 Résultat des simulations et statistique d'erreur



**Figure 4.10 : Reconnaissance faciale réussie**

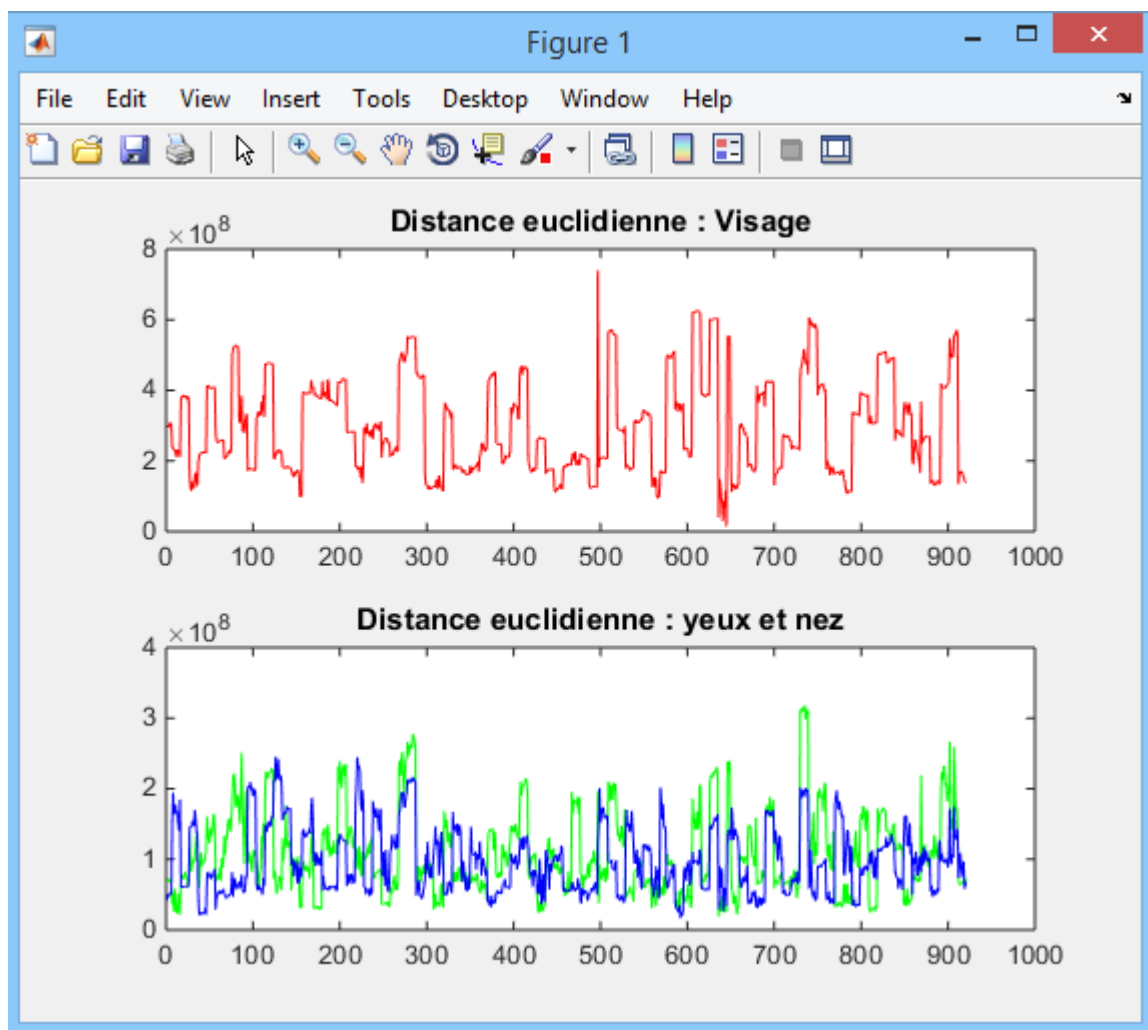
On a chargé ici une image quelconque dans l'une des dossiers tests. On a lancé la reconnaissance, le visage est reconnu, dont le nom est « mberdo » (le nom correspond au nom du dossier de la personne). La distance minimale est de  $2.44 \times 10^7$  qui est inférieure au seuil de décision qu'on a pris :  $3,6 \times 10^7$ . Dans le cas où cette distance minimale sera supérieure à  $3,6 \times 10^7$ , la personne sera considérée comme inconnue.

Pour avoir une statistique de reconnaissance, on a fait des tests sur des visages qui n'appartiennent pas à la base, des visages qui n'ont qu'une seule image dans la base, et ceux qui n'ont que deux ou trois et aussi ceux qui ont plus de cinq images dans la base.

On a récupéré les distances minimales et qui sont représentées dans le tableau 4.01.

0 image	1 image	2 à 3 images	Plus de 5 images
4.63	5.21	1.01	1.40
10.06	1.24	2.89	2.95
4.95	2.34	3.26	1.17
6.06		6.86	0.90

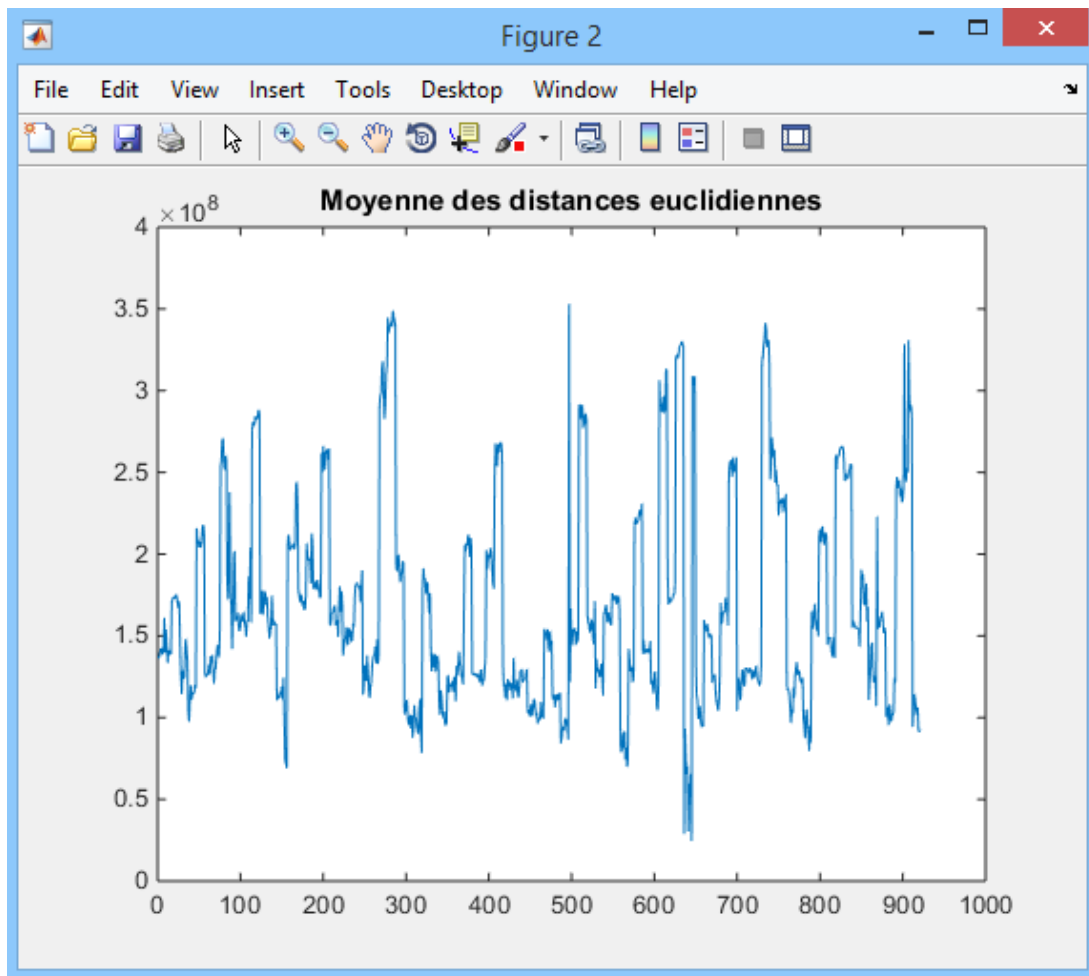
**Tableau 4.01 :** *Statistique des distances euclidiennes ( $\times 10^7$ )*



**Figure 4.11 :** *Distances euclidiennes des visages, des yeux et du nez*

Ce graphe illustré dans la figure 4.11 représente l'ensemble des distances euclidiennes des visages, des yeux et du nez. L'axe des abscisses représente l'indexation des images et l'axe des ordonnées représente les distances entre l'image test et les images indexées.

Puis, on a calculé les moyennes de ces trois distances euclidiennes et le graphe obtenu est présenté dans la figure 4.12. On peut voir le minimum des distances euclidiennes dans ces graphes.



**Figure 4.12 :** *Distances euclidiennes moyennes*

## 4.6 Conclusion

Ce chapitre a permis d'en savoir plus sur le logiciel Matlab avec les fonctions les plus utilisées dans le domaine de calcul matriciel et vectoriel, c'est-à-dire quelques opérations de base. Certaines fonctions utiles à l'application sont aussi présentées. Purement développé par Matlab, l'algorithme fait la détection du visage, des yeux et du nez ; et fait la reconnaissance de ces derniers. En utilisant une base de données de visage Face94, on a pu tester la robustesse du programme pour 100 sujets ; et on peut dire que celui-ci est bien fiable et acceptable.

## CONCLUSION GENERALE

La biométrie est une technologie qui reste toujours un domaine de recherche, car la finalité attendue n'est pas encore satisfaisante. Chaque système présente soit des inconvénients sur la taille de donnée trop grande, soit le temps de traitement trop lent, soit la précision qui n'est pas très convaincante. Mais du moins, certains systèmes biométriques sont plus acceptables que d'autres selon l'application concernée.

Dans ce mémoire, nous avons choisi le visage comme caractéristique biométrique ; car parmi tous les traits utilisés, les caractéristiques faciales marquent le plus de compatibilité dans plusieurs domaines. La reconnaissance faciale se compose des deux phases dans son processus : la phase d'apprentissage et la phase de test. On a vu aussi les trois différents algorithmes de reconnaissance faciale et parmi eux, la méthode hybride qui exploite à la fois les avantages de la méthode globale et de la méthode locale pour avoir une meilleure performance.

L'Analyse en Composante Principale est un algorithme à méthode globale, mais d'un autre côté, on a aussi utilisé ce système d'analyse de donnée sur les traits locaux du visage, qui est l'algorithme ACP modulaire. Dans notre système, on utilise à la fois l'eigenfaces normal avec l'eigenfaces modulaire pour avoir un algorithme à méthode hybride. D'après quelques tests pendant la conception de notre logiciel, la combinaison « visage, yeux, nez » donne le meilleur résultat.

En utilisant une base de données de visage standard, avec 100 individus, 10 images de visage au maximum pour chacun. Notre algorithme donne un très bon résultat avec un temps d'apprentissage lent, mais une reconnaissance assez rapide et fiable pour 100 sujets.

Bref, cet algorithme est très prometteur si on veut mettre au point un système de vérification biométrique à l'accès d'une enceinte partagée, par exemple : département, salle d'étude, médiathèque privée, etc. La possibilité d'utilisation de notre système est très vaste, c'est pour cela que créer le modèle multiplateforme sera indispensable, en Android, en IOS,... L'intégration sur des projets domotique est aussi envisageable... En un mot, l'exploitation totale de cet algorithme reste un projet en cours.

## ANNEXE 1

### OPERATIONS SUR LES MATRICES ET VECTEURS

#### A1.1 Opérations sur les matrices

##### A1.1.1 Transposition

On peut écrire des données matricielles en inversant le rôle des lignes et des colonnes. On dit qu'on peut *transposer* la matrice. La matrice transposée se note  $A^T$  (on lit « A transpose » ou « A transposée ») :

$$\begin{array}{ll} \text{si} & A = A_{(3 \times 4)} = \begin{bmatrix} 2 & 5 & 10 & 20 \\ 1 & 2 & 3 & 4 \\ 6 & 1 & 3 & 10 \end{bmatrix} \\ \text{alors} & A^T = A_{(4 \times 3)}^T = \begin{bmatrix} 2 & 1 & 6 \\ 5 & 2 & 3 \\ 10 & 3 & 3 \\ 20 & 4 & 10 \end{bmatrix} \end{array} \quad (\text{A1.01})$$

Avec  $(A^T)^T = A$

##### A1.1.2 Addition (somme) de matrices

L'intérêt majeur des matrices vient de ce qu'elles généralisent les opérations familières des nombres réels. Certaines de ces opérations se généralisent de manière assez évidente, d'autres de manière plus sophistiquée.

Deux matrices ayant le même nombre de lignes et de colonnes peuvent s'additionner terme à terme.

La matrice résultante est de mêmes dimensions que les deux matrices additionnées.

De manière plus générale :

$$A + B = \begin{bmatrix} a_{1,1} + b_{1,1} & a_{1,2} + b_{1,2} & \dots & a_{1,j} + b_{1,j} \\ a_{2,1} + b_{2,1} & a_{2,2} + b_{2,2} & \dots & a_{1,j} + b_{1,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} + b_{i,1} & a_{i,2} + b_{i,2} & \dots & a_{i,j} + b_{i,j} \end{bmatrix} \quad (\text{A1.02})$$

Comme pour les nombres habituels, l'addition de matrices est commutative c'est-à-dire  $A+B = B+A$  ; et associative, c'est-à-dire  $A + (B+C) = (A+B) + C$ .

##### A1.1.3 Multiplication d'une matrice par un scalaire

On peut multiplier tous les éléments d'une matrice par un même nombre. Pour différencier les matrices des nombres classiques, on appelle ces derniers des « *nombres scalaires* » ou, pour abrégé,

des *scalaires*. Pour multiplier une matrice par un scalaire, on multiplie chaque nombre de la matrice par ce scalaire. Par exemple :

$$10 \times B = 10 \times \begin{bmatrix} 3 & 4 & 5 & 6 \\ 2 & 4 & 6 & 8 \\ 1 & 2 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 10 \times 3 & 10 \times 4 & 10 \times 5 & 10 \times 6 \\ 10 \times 2 & 10 \times 4 & 10 \times 6 & 10 \times 8 \\ 10 \times 1 & 10 \times 2 & 10 \times 3 & 10 \times 5 \end{bmatrix} \quad (A1.03)$$

#### ***A1.1.4 Soustraction***

Pour soustraire une matrice d'une autre matrice, il suffit de multiplier la seconde matrice par -1 et ensuite de l'additionner à la première matrice.

#### ***A1.1.5 Produit de Hadamard de deux matrices***

Pour généraliser l'opération de multiplication aux matrices, la première approche définit la multiplication de deux matrices de manières similaires à l'addition. On appelle cette multiplication produit terme par terme ou produit de Hadamard [16]. Tout comme l'addition, le produit de Hadamard n'est défini que pour des matrices de mêmes dimensions. On dénote le produit de Hadamard avec le signe  $\odot$ . Ainsi, le produit de Hadamard de A et B se note  $A \odot B$ . et se définit comme :

$$A \odot B = [a_{i,j} \times b_{i,j}]$$

$$A \odot B = \begin{bmatrix} a_{1,1} \times b_{1,1} & a_{1,2} \times b_{1,2} & \dots & a_{1,j} \times b_{1,j} \\ a_{2,1} \times b_{2,1} & a_{2,2} \times b_{2,2} & \dots & a_{1,j} \times b_{1,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} \times b_{i,1} & a_{i,2} \times b_{i,2} & \dots & a_{i,j} \times b_{i,j} \end{bmatrix} \quad (A1.04)$$

Par exemple avec les matrices A et B suivantes :

$$A \odot B = \begin{bmatrix} 2 \times 3 & 5 \times 4 & 10 \times 5 & 20 \times 6 \\ 1 \times 2 & 2 \times 4 & 3 \times 6 & 4 \times 8 \\ 6 \times 1 & 1 \times 2 & 3 \times 3 & 10 \times 5 \end{bmatrix} = \begin{bmatrix} 6 & 20 & 50 & 120 \\ 2 & 8 & 18 & 32 \\ 6 & 2 & 9 & 50 \end{bmatrix}$$

#### ***A1.1.6 Division de Hadamard***

La division de Hadamard se définit de manière analogue au produit de Hadamard. C'est-à-dire comme la division terme à terme de deux matrices de mêmes dimensions [16]. Elle se note  $\oslash$ . Par exemple, avec les matrices :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ et } B = \begin{bmatrix} 5 & 10 & 6 \\ 12 & 25 & 6 \end{bmatrix}$$

On obtient :

$$B \oslash A = B = \begin{bmatrix} \frac{5}{1} & \frac{10}{2} & \frac{6}{3} \\ \frac{12}{4} & \frac{25}{5} & \frac{6}{6} \end{bmatrix} = \begin{bmatrix} 5 & 5 & 2 \\ 3 & 5 & 1 \end{bmatrix} \quad (\text{A1.05})$$

La division de Hadamard n'est pas définie lorsque l'un des termes de la deuxième matrice est nul.

#### ***A1.1.7 Produit (standard ou de Cayley) de deux matrices***

Le produit de Hadamard, même s'il semble être une bonne idée, ne correspond pas au produit habituel de deux matrices. La raison essentielle découle de l'utilisation de matrices pour résoudre des systèmes d'équations linéaires. Dans ce contexte, il est plus naturel de définir le produit de deux matrices de façon différente. Le produit le plus courant pour les matrices est appelé *produit standard* ou parfois aussi produit de *Cayley*, ou simplement *produit* (lorsque le produit n'est pas spécifié, il s'agit du produit standard) [16].

Le produit de deux matrices n'est défini que dans le cas particulier où le nombre de colonnes de la première matrice est égal au nombre de lignes de la deuxième matrice. La matrice produite aura le nombre de lignes de la première matrice et le nombre de colonnes de la deuxième matrice. On dit dans ce cas que les deux matrices sont confortables.

$$A_{I \times J} \times B_{J \times K} = C_{I \times K} \quad (\text{A1.06})$$

La matrice C aura pour terme générique  $c_{i,k}$  qui s'obtient comme suit :

$$c_{i,k} = \sum_{j=1}^j a_{i,j} \times b_{j,k} \quad (\text{A1.07})$$

Exemple :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ et } B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

alors  $AB = C = [c_{i,k}]$

$$C = \begin{bmatrix} 1 \times 1 + 2 \times 3 + 3 \times 5 & 1 \times 2 + 2 \times 4 + 3 \times 6 \\ 4 \times 1 + 5 \times 3 + 6 \times 5 & 4 \times 2 + 5 \times 4 + 6 \times 6 \end{bmatrix} = \begin{bmatrix} 22 & 28 \\ 49 & 64 \end{bmatrix}$$



Le produit matriciel est associatif et distributif. Les produits AB et BA ne peuvent exister simultanément que si A et B sont de mêmes dimensions. Mais contrairement au produit des nombres habituels, le produit matriciel n'est pas commutatif.

#### **A1.1.8 Produit exotique : Kronecker**

Il existe encore une autre manière de définir la notion de produit pour des matrices. Il s'agit du produit de Kronecker aussi appelé produit direct de deux matrices ou produit tensoriel ou parfois encore produit de Zehfuss. Il se note avec le symbole  $\otimes$  [16].

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,j}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,j}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1}B & a_{i,2}B & \dots & a_{i,j}B \end{bmatrix} \quad (\text{A1.08})$$

Par exemple, avec les matrices A et B suivantes :

$$A = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{ et } B = \begin{bmatrix} 6 & 7 \\ 8 & 9 \end{bmatrix}$$

On obtient :

$$A \otimes B = \begin{bmatrix} 6 & 7 & 12 & 14 & 18 & 21 \\ 8 & 9 & 16 & 18 & 24 & 27 \end{bmatrix}$$

### **A1.2 Opérations sur les vecteurs**

Il existe plusieurs opérations que les matrices utilisent et les vecteurs aussi, en respectant les mêmes propriétés. On considère seulement que le vecteur est une matrice à une colonne.

#### **A1.2.1 Produit scalaire de 2 vecteurs**

On définit le *produit scalaire* (aussi appelé parfois produit interne) de deux vecteurs a et b comme :

$$a^T \times b = a^T b = \sum_{i=1}^I a_i \times b_i \quad (\text{A1.09})$$

#### **A1.2.2 Produit externe de deux vecteurs**

On peut associer une matrice à tout couple de vecteurs avec l'opération  $ab^T$ . On appelle cette opération le *produit externe* de deux vecteurs.

D'une façon générale, le produit externe de deux vecteurs a et b s'obtient comme :

$$ab^T = \begin{bmatrix} a_1 \times b_1 & a_1 \times b_i & \dots & a_1 \times b_j \\ \vdots & \vdots & \ddots & \vdots \\ a_i \times b_1 & a_i \times b_i & \dots & a_i \times b_j \\ \vdots & \vdots & \ddots & \vdots \\ a_j \times b_1 & a_j \times b_i & \dots & a_j \times b_j \end{bmatrix} \quad (\text{A1.10})$$

### A1.2.3 Opérateur VEC

Il arrive souvent que l'on ait besoin d'écrire une matrice sous forme de vecteur. L'opération qui réécrit la matrice A en un vecteur dans lequel les colonnes de A sont écrites les unes après les autres, se note  $\text{vec}\{A\}$  [16]. Ainsi à partir de :

$$A = [a_{i,j}] = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,j} \\ a_{2,1} & a_{2,2} & \dots & a_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} & a_{i,2} & \dots & a_{i,j} \end{bmatrix} = [a_1, \dots, a_i, \dots, a_j]$$

(Avec  $a_j$  étant la j-ième colonne de A). On obtient :

$$\text{vec}\{A\} = \begin{bmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{i,1} \\ \vdots \\ a_{j,1} \\ a_{1,2} \\ \vdots \\ a_{1,j} \\ \vdots \\ a_{i,j} \end{bmatrix} \quad (\text{A1.11})$$

## **ANNEXE 2**

### **BASE DE DONNEES DE VISAGE**

Lors de l'évaluation d'un algorithme, il est recommandé d'utiliser un ensemble de données de test standard pour pouvoir comparer directement les résultats. Bien qu'il existe de nombreuses bases de données en cours d'utilisation actuellement, le choix d'une base de données appropriée à utiliser doit être fait sur la base de la tâche donnée (vieillesse, expressions, éclairage, etc.). Une autre façon est de choisir l'ensemble spécifique selon la propriété des données à tester (par exemple, comment l'algorithme se comporte sur des images avec des changements d'éclairage ou des images avec différentes expressions faciales). Si, d'autre part, un algorithme doit être formé avec plus d'images par classe (comme ADL), la base de données de visage Yale est probablement plus appropriée que FERET.

On a pris dans [27] quelques ensembles de données de visage souvent utilisées par les chercheurs:

#### **A2.1 The Color FERET Database, USA**

Le programme FERET a pour but d'établir une grande base de données d'images du visage, qui ont été recueillies de façon indépendante par les développeurs d'algorithmes. Dr. Harry Wechsler à l'Université George Mason a été choisi pour diriger la collecte de cette base de données. La collection de bases de données a été un effort de collaboration entre le Dr Wechsler et le Dr Phillips. Les images ont été recueillies dans un milieu semi-contrôlé. Pour maintenir un degré de cohérence dans la base de données, la même configuration physique a été utilisée dans chaque séance photo. Parce que l'équipement devait être remonté pour chaque session, il y avait une certaine variation mineure dans les images recueillies à des dates différentes. La base de données FERET a été recueillie en 15 sessions entre août 1993 et juillet 1996. La base de données contient 1.564 ensembles d'images pour un total de 14.126 images qui comprend 1.199 individus et 365 ensembles en double d'images. Un ensemble en double est un second ensemble d'images d'une personne déjà dans la base de données et a généralement été pris un autre jour. Pour certaines personnes, plus de deux ans se sont écoulés entre les premières et dernières séances, avec quelques sujets photographiés à plusieurs reprises. Ce laps de temps est important, car il a permis aux chercheurs d'étudier, pour la première fois, les changements dans l'apparence d'un sujet qui se produisent au cours d'une année.

#### **A2.2 SCface - Surveillance Cameras Face Database**

SCface est une base de données d'images statiques de visages humains. Les images ont été prises dans l'environnement intérieur non contrôlé à l'aide de cinq caméras de surveillance vidéo de

différentes qualités. La base de données contient 4.160 images statiques de 130 sujets. Les images des caméras de différentes qualités imitent les conditions du monde réel et permettent de tester des algorithmes de reconnaissance de visage robuste, mettant l'accent sur les différentes forces de l'ordre et l'utilisation de la surveillance des scénarios. La base de données SCface est disponible gratuitement à la communauté de recherche.

### **A2.3 SCfaceDB Landmarks**

La base de données est composée de 21 points de repère du visage (à partir de 4.160 images de visage) de 130 utilisateurs annotés manuellement par un opérateur humain.

### **A2.4 Multi-PIE**

Une relation étroite existe entre l'avancement des algorithmes de reconnaissance de visage et la disponibilité des bases de données de visage avec des facteurs variables qui affectent l'apparence du visage d'une manière contrôlée. La base de données PIE, recueillie à l'Université Carnegie Mellon en 2000, a été très influente dans la promotion de la recherche dans la reconnaissance des visages avec changement de pose et d'éclairage. Malgré son succès, la base de données PIE présente plusieurs inconvénients : un nombre limité de sujets, une session d'enregistrement unique et seulement quelques expressions capturées. Pour répondre à ces demandes, les chercheurs de l'Université Carnegie Mellon ont recueilli la base de données Multi-PIE. Il contient 337 sujets, capturés au moins de 15 points de vue différents et 19 conditions d'éclairage, à quatre sessions d'enregistrement pour un total de plus de 750.000 images.

### **A2.5 The Yale Face Database A et B**

La base de données Yale Face A contient 165 images en niveaux de gris au format GIF de 15 personnes. Il y a 11 images par sujet, une par expression du visage suivant les différentes configurations : lumière centrée, heureux, avec lunettes, lumière à gauche, sans lunettes, normales, lumière à droite, triste, endormi, surpris, et clin d'œil.

La version B contient 5.760 images de 10 sujets, chacun vu sous 576 conditions de visualisation (9 poses x 64 conditions d'éclairage). Pour chaque sujet, une image avec un éclairage ambiant a également été capturée dans une pose particulière.

### **A2.6 Project - Face In Action (FIA) Face Video Database, AMP, CMU**

La séance de capture imite les applications du monde réel. Par exemple, quand une personne passe par un check-point de l'aéroport, six caméras capturent les visages humains sous trois angles

différents. Trois des six caméras ont un plus petit focus, et les trois autres avec un plus grand focus. Les séances ont été planifiées pour capturer 200 sujets en trois sessions de périodes différentes. Pour une session, on fait à la fois des séances photo à l'extérieur et à l'intérieur. La pose et la variation d'expression successive de l'utilisateur forment des séquences vidéo.

#### **A2.7 AT&T "The Database of Faces" (ou "The ORL Database of Faces")**

Cette base de données dispose de dix images différentes chacune pour 40 sujets distincts. Pour certains sujets, les images ont été prises à des moments différents, variant l'éclairage, les expressions du visage (yeux ouverts/fermés, heureux/triste) et les détails du visage (avec lunettes/sans lunettes). Toutes les images des sujets ont été prises sur un fond homogène foncé, dans une position portrait frontale (avec certain mouvement latéral).

#### **A2.8 MIT-CBCL Face Recognition Database**

La base de données de reconnaissance faciale MIT-CBCL contient des images du visage de 10 sujets. Il fournit deux ensembles d'apprentissages : des images de haute résolution, contenant une photo frontale, demi-profil et profil ; et des images synthétiques (324 par sujets) rendues à partir de modèles 3D de tête des 10 personnes. Les modèles de tête ont été générés en ajustant un modèle d'images formées à haute résolution. Les modèles 3D ne sont pas inclus dans la base de données. L'ensemble de tests se compose de 200 images par sujet. Quelques variations d'éclairage, de la pose (jusqu'à environ 30 degrés de rotation en profondeur) ainsi que d'arrière-plan sont appliqués.

#### **A2.9 M2VTS Multimodal Face Database (Release 1.00)**

Cette base de données est composée de 37 visages différents et fournit cinq photos pour chaque personne. Ces images ont été prises à des intervalles d'une semaine. Au cours de chaque prise, les gens ont été invités à compter de '0' à '9' dans leur langue maternelle (la plupart des gens parlent français), faire tourner la tête de 0 à -90 degrés, encore une fois à 0, puis à +90 et retour à 0 degré. En outre, ils ont été invités à faire tourner la tête une fois de plus sans lunettes si en elles portent.

#### **A2.10 The AR Face Database, The Ohio State University, USA**

La base de visage AR contient 4.000 images couleur correspondant aux faces de 126 personnes (70 hommes et 56 femmes). Les images disposent des vues frontales avec différentes expressions faciales, de conditions d'éclairage et d'accessoires (lunettes de soleil et écharpe).

### **A2.11 CAS-PEAL Face Database**

La base de données de visage CAS-PEAL a été construite sous les sponsors du « National Hi-Tech Program and ISVISION ». Les objectifs pour créer la base de données de visage PEAL est de : fournir aux chercheurs du monde entier de la communauté FR (Face Recognition) une base de données de visage chinois à grande échelle pour l'apprentissage et l'évaluation de leurs algorithmes ; faciliter le développement de la reconnaissance faciale en fournissant des images de visage à grande échelle avec différentes sources de variations, en particulier : Pose, Expression, Accessoires et Eclairage (PEAL) ; faire avancer les technologies de pointe de la reconnaissance faciale qui visent surtout des applications pratiques pour les Orientaux.

### **A2.12 VALID Database**

Dans le but de faciliter le développement des systèmes robuste de reconnaissance vocale, faciale et multimodale des personnes, la base de données VALID a été acquise dans un « monde réel » : scénario de bureau bruyant avec aucun contrôle sur l'éclairage et les bruits acoustiques. La base de données se compose de cinq séances d'enregistrement de 106 sujets sur une période d'un mois. Une séance est enregistrée dans un studio avec un éclairage contrôlé et sans bruit de fond, les quatre autres séances sont enregistrées dans des scénarios de type bureau. La base de données contient des images JPEG non compressées à une résolution de 720x576 pixels.

### **A2.13 Natural Visible and Infrared facial Expression database (USTC-NVIE)**

La base de données contient des expressions spontanées et poses de plus de 100 sujets, enregistrées simultanément par une caméra normale et une caméra à infrarouge thermique, avec un éclairage fourni à partir de trois directions différentes. La base de données contient également des images avec des changements de pose, d'expression et des visages avec et sans lunettes.

### **A2.14 FEI Face Database**

La base de données de visage FEI est une base de données de visage brésilien qui contient un ensemble d'images de visage prises entre juin 2005 et mars 2006 au Laboratoire d'Intelligence Artificielle de FEI à Sao Bernardo do Campo, Sao Paulo, Brésil. Il y a 14 images pour chacun des 200 individus, soit un total de 2.800 images. Toutes les images sont colorées et prises sur un fond blanc homogène dans une position frontale droite avec des profils en rotation allant jusqu'à environ 180 degrés. L'échelle peut varier d'environ 10% par rapport à la taille d'origine de chaque image qui est de 640x480 pixels. Tous les visages sont principalement représentés par les étudiants et le

personnel de la FEI, entre 19 et 40 ans avec une apparence distincte, coiffure, et orne. Le nombre de sujets masculins et féminins sont exactement les mêmes et égal à 100.

Il existe encore plus de 75 bases de données de visage utilisé pour le développement de système de reconnaissance faciale, avec chacun ses propres caractéristiques.

### ANNEXE 3

#### CODE DE L'ACP SOUS MATLAB

Voici la fonction qui calcule l'ACP des bases d'images qui sont déjà vectorisées et concaténées pour avoir la matrice T :

```
function [m, A, Eigenfaces] = EigenfaceCore(T, Type)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Calculer l'image moyenne
m = mean(T,2); % Ceci suit la formule m = (1/P)*sum(Tj's)      (j = 1 : P)
Train_Number = size(T,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Calculer l'écart de chaque image sur l'image moyenne
A = [];
for i = 1 : Train_Number
    temp = double(T(:,i)) - m; % Calculer la différence pour chaque image de
    l'apprentissage Ai = Ti - m
    A = [A temp]; % enregistrer tous les images centrées
end
L = A'*A; % L est le substitut de la matrice de covariance C=A*A'.
[V D] = eig(L); % Les éléments diagonaux de D sont les valeurs propres
% Et une matrice V dont les colonnes sont les vecteurs propres de L=A'*A et
C=A*A'.
L_eig_vec = [];
for i = 1 : size(V,2)
    L_eig_vec = [L_eig_vec V(:,i)];
end
Eigenfaces = A * L_eig_vec; % A: vecteurs de l'image centrée
```

On peut voir ici la fonction calculant la distance entre l'image test et les images de la base, c'est la fonction de reconnaissance ou fonction de test.

```
function [Euc_dist] = Recognition(TestImage, m, A, Eigenfaces)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% fonction de test %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% distance entre TestImage et les bases de visages %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ProjectedImages = [];
Train_Number = size(Eigenfaces,2);
```



```

for i = 1 : Train_Number
    temp = Eigenfaces'*A(:,i); % Projection d'images centrées dans facespace
    ProjectedImages = [ProjectedImages temp];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Extraction des caractéristiques ACP de l'image test
InputImage = imread(TestImage);

temp = rgb2gray(InputImage);

[irow icol] = size(temp);
InImage = reshape(temp',irow*icol,1);
Difference = double(InImage)-m; % image test centrée
ProjectedTestImage = Eigenfaces'*Difference; % Le vecteur caractéristique de
l'image testé

% Distance euclidien
Euc_dist = [];
for i = 1 : Train_Number
    q = ProjectedImages(:,i);
    temp = ( norm( ProjectedTestImage - q ) );%^2;
    Euc_dist = [Euc_dist temp];
end

```

## BIBLIOGRAPHIE

- [1] L. Mansoura, « *Identification des Visage Humains par Réseaux de Neurones* », Mémoire de fin d'étude, Dép. électronique, option : Traitement d'image, Université de Batna, A.U : 2012-2013.
- [2] B. Sofiane, « *Détection et identification de personne par méthode biométrique* », Mémoire de magister en Electronique, Option : Télédétection, Université Mouloud MAMMERI et TIZI-OUZOU.
- [3] P. Viet Anh, « *Développement d'un module de segmentation pour un système de reconnaissance biométrique basé sur l'iris* », mémoire de fin d'étude, Institut National des Télécommunication, France, Novembre 2008.
- [4] R. Giot, « *Apprentissage incrémental pour les systèmes biométriques* », Avril 2011
- [5] G. Dreyfus et al. « *Apprentissage statique* », Eyrolles, 3ème Edition.
- [6] S.Z. Li, A.K. Jain, « *Handbook of Face Recognition* », Springer, United States of America.
- [7] « *Les contrôles d'accès physique par biométrie* », CLUSIF, Juin 2003
- [8] « *Reconnaissance vocale* », Recherche et technologies, Multitel.
- [9] T. Rawlinson et al., « *Principles and Methods for Face Recognition and Face Modelling* » Dep. of Computer Science, University of Warwick, UK. Février 2009
- [10] Y. Sabrina, « *Un modèle basé Template Matching/Réseau de Neurones pour la Reconnaissance des visages* », Mémoire Magister, Université Badji Makhtar, A.U : 2006
- [11] M. A. Fodda, « *Détection et reconnaissance de visage* », Projet de fin d'études, ESPRIT, Octobre 2010.
- [12] P. Sinha, « *Processing and Recognizion 3D format* », PhD thesis, Massachusetts Inst. of Technology, 1995.
- [13] S. Guerfi Ababsa, « *Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D* », Thèse de doctorat, UNIVERSITE D'EVRY VAL D'ESSONNE, octobre 2008.

- [14] R. Jourani, « *Reconnaissance de visage* », Mémoire de fin d'études, Université Mohammend V-Agdal, Novembre 2006.
- [15] B. Abdessettar, S. Fathi, « *Extraction des caractéristiques pour l'analyse biométrique d'un visage* », Université Kasdi MERBAH OUARGLA, A.U : 2013-1204
- [16] H. Abdi, D. Valentin, « *Mathématique pour les sciences cognitives* », Presses universitaires de Grenoble, 2006.
- [17] A. Morineau, « *ACP-Analyse en Composantes Principales* », deenov.
- [18] S. Bazeille, « *Vision sous-marine monoculaire pour la reconnaissance d'objets* », Thèse de docteur en sciences, Université de Bretagne Occidentale, A.U : Octobre 2008.
- [19] B. Abdelouahid, N. Hamza, « *Reconnaissance de visage par réseau de neurones* », Laboratoire de recherche Informatique et Télécommunications, UNIVERSITÉ MOHAMMED V, A.U : 2014-2015
- [20] N. Martaj, M. Mokhtari, « *MATLAB R2009, SIMULINK et STATEFLOW pour Ingénieurs, Chercheurs et Etudiants* », Springer.
- [21] H. Le-Huy, « *Introduction à MATLAB et Simulink* », Université Laval, Québec, 1998.
- [22] M. Van Wambeke, « *Reconnaissance et suivi de visages et implémentation en robotique temps-réel* », Mémoire de fin d'études Master Ingénieur Civil en Génie Biomédical, Université Catholique de Louvain, A.U : 2009-2010.
- [23] Y. Q. Wang, « *An Analysis of the Viola-Jones Face Detection Algorithm* », Image Processing On Line, 2014.
- [24] « Larousse.fr : encyclopédie et dictionnaires », <http://www.larousse.fr/>, janvier 2016
- [25] « L'Internaute : actualité, loisirs, culture et découvertes... », <http://www.linternaute.com/>, février 2016
- [26] Help Matlab, novembre 2015
- [27] « University of Essex », <http://cswww.essex.ac.uk>, mars 2016

## **FICHE DE RENSEIGNEMENTS**

**Nom :** RANDRIAMAHANDRY  
**Prénom :** Vonjinirina Eric  
**Adresse :** Lot IVA 7 Antanetisoa Ankaraobato  
Antananarivo 102  
Madagascar  
**Tel :** +261 32 92 833 62  
**Email :** randriamahandryeric@hotmail.fr



**Titre de mémoire :**

**RECONNAISSANCE FACIALE PAR METHODE ACP HYBRIDE**

**Nombres de pages : 75**

**Nombres de tableaux : 3**

**Nombre de figures : 42**

**Directeur de mémoire :**

**Nom :** ANDRIANTSILAVO  
**Prénom :** Haja Samiarivonjy  
**Tel :** +261 33 14 223 23

## RESUME

Reconnaître une personne par des traits distinctifs, c'est l'idée principale des systèmes biométriques. La biométrie est une technologie qui depuis sa création, n'arrête pas d'évoluer, mais le système idéal reste encore à venir. L'un des biométries fondamentales les plus populaires est la reconnaissance faciale, parce que c'est un système naturel, non intrusif et facile à utiliser. L'ACP ou eigenfaces est un algorithme de reconnaissance du visage à méthode globale, qui analyse les composantes principales de la représentation matricielle du visage. En l'utilisant avec sa variante à méthode locale, l'ACP modulaire ; on obtient un algorithme hybride avec une meilleure performance. Le logiciel qu'on a conçu avec ce nouvel algorithme renvoie un résultat très satisfaisant, ce qui lui permettrait d'être utilisé en pratique, dans des vérifications de contrôles d'accès quotidiens. Programmer sur Matlab, le test du logiciel a atteint une authentification jusqu'à 100 sujets.

**Mots clés :** Reconnaissance faciale, ACP, détection visage, Matlab, méthode hybride.

## ABSTRACT

To recognize a person by distinctive features, it is the main idea of the systems biometrics. The biometry is a technology that since its creation, don't stop evolving, but the ideal system remained to come again. One of the most popular fundamental biometrics is the face recognition, because it is a natural system, no intrusive and easy to use. The PCA or eigenfaces is an algorithm of face recognition to global method, which analyzes the principal components of the matrix representation of the face. While using it with its variant to local method, the modular PCA; one gets a hybrid algorithm with a better performance. The software that one conceived with this new algorithm sends back a very satisfactory result, what would allow him to be used in practice, in verifications of controls of daily accesses. To program on Matlab, the test of the software reached an authentication until 100 users.

**Keywords:** Face recognition, PCA, Face detection, Matlab, hybrid algorithm.