

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB

Faculté des Sciences
Département Informatique



**Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.**
Option : Système d'Information

Sujet :

**Etude comparative entre la DCT-
PCA et la DCT-DPCA dans la
reconnaissance de visage**

Présenté par : BENNAIDJA Samir **Promoteur :** Mr Bengherabi M.
DRIOUECHE El Mehdi **Encadreur :** Mr Harizi F.

Organisme d'accueil : Centre de Développement des Technologies Avancées.

Soutenu devant le jury composé de :

M^r MAHIEDDINE
M^r HAMMOUDA
M^r CHERIF ZAHAR

Président
Examineur
Examineur

- 2006 / 2007 -

MIG-004-176.1

REMERCIEMENTS

Nous tenons à remercier MM. Belgherabi Messaoud et Harizi Farid pour la proposition du sujet, le suivi du travail et pour leurs nombreux conseils et leur soutien moral.

Sans oublier l'apport de Mme S. Benbrahim qui a bien voulu lire notre travail et le corriger.

Nous tenons également à remercier les étudiants du département d'informatique qui ont participé à l'élaboration de la base de données Blida Visage.

Que les membres du jury trouvent ici le témoignage de notre reconnaissance pour avoir bien voulu évaluer et juger notre travail.

Nous tenons à remercier tous les enseignants que nous avons eus durant tout notre cursus et plus particulièrement les enseignants du département d'informatique sans qui nous ne serions pas là.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à l'élaboration de notre travail.

TABLE DES MATIERES

Introduction Générale	- 11
PARTIE 1:BIOMETRIE ET RECONNAISSANCE DE VISAGE	
Chapitre I : Introduction à la Biométrie	
Introduction	- 16
I.2. Méthode d'authentification biométrique	
I.3. Présentation des différents systèmes biométriques	- 17
I.3.1. Systèmes morphologiques	
I.3.1.1. Empreintes digitales	
I.3.1.2. Géométrie de la main/du doigt	- 18
I.3.1.3. Iris	
I.3.1.4. Rétine	- 19
I.3.1.5. Visage	
I.3.1.6. Reconnaissance vocale	
I.3.2. Systèmes comportementaux	- 20
I.3.2.1. Dynamique des signatures	
I.3.2.2. Dynamique de la frappe au clavier	
I.4. État du marché de la biométrie	- 21
Conclusion	- 22
Chapitre II: Notions de base sur le traitement d'images	
II.1. Définition générale	- 24
II.2. Définition d'une image numérique	
II.3. Caractéristiques d'une image numérique	- 25
II.3.1. Pixel	
II.3.2. Résolution	
II.3.3. Dimension	
II.3.4. Représentation des couleurs	- 26
II. 3.5. Image en niveaux de gris	
II.3.6. Histogramme d'une image	- 27
II.3.7.Contours et textures	
II.3.8. Luminance	
II.3.9. Contraste	
II.3.10. Convolution	- 28
II.4. Prétraitement	
II.4.1.Traitement à base d'histogramme	
II.4.1.1. Normalisation	
II.4.1.2. Egalisation	- 29
II.4.2. Filtrage	- 30
II.4.2.1.Filtres linéaires	
II.4.2.2. Filtres non linéaires	- 32
II.4.2.3. Filtres de détection de contours	- 33
Conclusion	- 34
PARTIE 2 : ETAT DE L'ART DES DIFFERENTS SYSTEMES DE RECONNAISSANCE	
Chapitre III: Système de reconnaissance de visage	
Introduction	- 37
III.1. Système de reconnaissance biométrique	- 38
III.2. Pourquoi le visage?	

III.3. Processus de reconnaissance de visage	-	-	-	-	-	-	-	-	39
III.3.1. Acquisition	-	-	-	-	-	-	-	-	40
III.3.2. Prétraitement	-	-	-	-	-	-	-	-	41
III.3.3. L'extraction de paramètres									
III.3.3.1. Transformation en Cosinus Discrète (DCT)									
III.3.3.2. Analyse en Composante Principale (PCA ou Eigen Faces)									
III.3.3.3. Analyse Discriminante Linéaire (LDA)	-	-	-	-	-	-	-	-	42
III.3.3.4. Machine à Vecteurs de Support (SVM)									43
III.3.3.5. Réseaux de neurones	-	-	-	-	-	-	-	-	44
III.3.3.6. Modèle surfacique du visage (3D)									
III.3.4. Test									
Conclusion	-	-	-	-	-	-	-	-	45
Chapitre IV: Transformée en Cosinus Discrète(DCT)									
Introduction	-	-	-	-	-	-	-	-	47
IV.1. Formule mathématique de DCT									
IV.2. Les caractéristiques de la DCT	-	-	-	-	-	-	-	-	48
IV.3. DCT vs KLT	-	-	-	-	-	-	-	-	50
IV.4. DCT vs DFT	-	-	-	-	-	-	-	-	51
IV.5. Types de DCT									
IV.5.1. La DCT par bloc 8x8									
IV.5.1.1. Définition									
IV.5.1.2. La sélection des coefficients	-	-	-	-	-	-	-	-	52
IV.5.1.3. Chevauchement des blocs 8x8									
IV.5.2. La DCT Globale	-	-	-	-	-	-	-	-	53
IV.5.2.1. Définition									
IV.5.2.2. Sélection des coefficients									
IV.6. La DCT dans la reconnaissance de visage	-	-	-	-	-	-	-	-	54
Conclusion	-	-	-	-	-	-	-	-	55
Chapitre V: Analyse en Composantes Principales (ACP)									
Généralités	-	-	-	-	-	-	-	-	57
V.1. Principe des PCA	-	-	-	-	-	-	-	-	58
V.2. Quelques propriétés des PCA									59
V.2.1. Concentration de l'information et réduction de dimensionnalité									
V.2.2. Indépendance de l'information	-	-	-	-	-	-	-	-	60
V.2.3. PCA et direction de la variance									
V.3. PCA Classique dans la reconnaissance de visage									
V.4. (2D) PCA dans la reconnaissance de visage	-	-	-	-	-	-	-	-	63
V.5. PCA dans le domaine DCT	-	-	-	-	-	-	-	-	65
Conclusion	-	-	-	-	-	-	-	-	66
Chapitre VI: Analyse Diagonale en Composantes Principales (Diag PCA)									
Introduction	-	-	-	-	-	-	-	-	68
VI.1. Définitions									
VI.2. Diagonale (2D) PCA	-	-	-	-	-	-	-	-	70
VI.3. Diagonale PCA	-	-	-	-	-	-	-	-	74
VI.4. Diagonale DCT	-	-	-	-	-	-	-	-	76
VI.5. Diagonale (2D) PCA dans le domaine DCT									
VI.6. Diagonale PCA dans le domaine DCT	-	-	-	-	-	-	-	-	77
Conclusion	-	-	-	-	-	-	-	-	78

PARTIE 3 : PARTIE PRATIQUE

Chapitre VII: Conception et réalisation

VII.1. Introduction	-	-	-	-	-	-	-	-	81
VII.2. Pourquoi UML									
VII.3. Démarche utilisée									
VII.4. Conception									
VII.4.1. Les acteurs	-	-	-	-	-	-	-	-	82
VII.4.2. Les cas d'utilisation									
VII.4.3. Diagramme de cas d'utilisation									
VII.4.4. Diagramme de séquence	-	-	-	-	-	-	-	-	83
VII.4.5. Diagramme de classe	-	-	-	-	-	-	-	-	86
VII.4.6. Diagramme de collaboration	-	-	-	-	-	-	-	-	92
VII.5. Réalisation	-	-	-	-	-	-	-	-	95
VII.5.1. Introduction									
VII.5.2. Les outils de développement									
VII.5.2.1. Microsoft Visual C++.NET									
VII.5.2.2. SQL SERVER 200	-	-	-	-	-	-	-	-	96
VII.5.3. Architecture du système de reconnaissance de visage									
VII.5.4. Les modules de l'application	-	-	-	-	-	-	-	-	98
VII.5.4.1. Forme principale									
VII.5.4.2. Base de données	-	-	-	-	-	-	-	-	99
VII.5.4.3. Apprentissage	-	-	-	-	-	-	-	-	100
VII.5.4.4. Test	-	-	-	-	-	-	-	-	101
VII.5.4.5. Acquisition et prétraitement									
Chapitre VIII: Etude comparative									
Introduction	-	-	-	-	-	-	-	-	104
VIII.2. Base de données utilisées									
VIII.2.1. ORL									
VIII.2.2. Blida Visage	-	-	-	-	-	-	-	-	105
VIII.3. Reconnaissance par PCA	-	-	-	-	-	-	-	-	106
VIII.3.1. PCA	-	-	-	-	-	-	-	-	107
VIII.3.1.1. Vecteurs PCA et taux de reconnaissance									
VIII.3.1.2. Luminance	-	-	-	-	-	-	-	-	108
VIII.3.2. DiaPCA (Diagonale PCA)	-	-	-	-	-	-	-	-	109
VIII.3.2.1. Vecteurs DiaPCA et taux de reconnaissance									
VIII.3.2.2. Luminance	-	-	-	-	-	-	-	-	110
VIII.3.3. (2D)PCA									111
VIII.3.3.1. Vecteurs (2D)PCA et taux de reconnaissance									
VIII.3.3.2. Luminance	-	-	-	-	-	-	-	-	112
VIII.3.4. Dia(2D)PCA	-	-	-	-	-	-	-	-	113
VIII.3.4.1. Vecteurs Dia(2D)PCA et taux de reconnaissance									
VIII.3.4.2. Luminance	-	-	-	-	-	-	-	-	114
VIII.3.5. Synthèse	-	-	-	-	-	-	-	-	115
VIII.4. Reconnaissance par DCT	-	-	-	-	-	-	-	-	116
VIII.4.1. Chevauchement									
VIII.4.2. Effets de la variation de luminance sur le taux de reconnaissance									
VIII.4.2.1. DCT Bloc 8x8	-	-	-	-	-	-	-	-	117
VIII.4.2.2. Diagonale DCT Bloc 8x8	-	-	-	-	-	-	-	-	118
VIII.4.2.3. DCT Globale (DCTG)	-	-	-	-	-	-	-	-	119
VIII.4.2.4. Diagonale DCTG (DiaDCTG)									

VIII.4.3. Synthèse	-	-	-	-	-	-	-	-	-	120
VIII.5. Reconnaissance par DCT-PCA										
VIII.5.1. DCT Bloc8x8-PCA										
VIII.5.1.1. Chevauchement										
VIII.5.1.2. Luminance	-	-	-	-	-	-	-	-	-	122
VIII.5.2. DCT Bloc8x8 Diag PCA	-	-	-	-	-	-	-	-	-	123
VIII.5.2.1. Vecteurs										
VIII.5.2.2. Luminance	-	-	-	-	-	-	-	-	-	124
VIII.5.3. DCTG (2D)PCA										
VIII.5.4.DCTG-Dia (2D)PCA	-	-	-	-	-	-	-	-	-	125
VIII.6. Synthèse										
Conclusion Générale	-	-	-	-	-	-	-	-	-	126
Annexe	-	-	-	-	-	-	-	-	-	128
Références sitographiques	-	-	-	-	-	-	-	-	-	149

LISTE DES TABLES

Table VII.1. La classe personne	-	-	-	-	-	88
Table VII.2. Les méthodes de la classe personne						
Table VII.3. La classe image	-	-	-	-	-	89
Table VII.4. Les méthodes de la classe image						
Table VII.5. La classe configuration						
Table VII.6. Les méthodes de la classe configuration						
Table VII.7. La classe modèle	-	-	-	-	-	90
Table VII.8. Format de la classe catégorie 1						
Table VII.9. Format de la classe catégorie 2	-	-	-	-	-	91
Table VII.10. Les méthodes de la classe model						
Table VII.11. Format de la classe approche						
Table VII.12. Format de la classe taux						
Table VII.13. Les méthodes de la classe taux	-	-	-	-	-	92

LISTE DES FIGURES

I.1 : Image d’empreinte	- 17
I.2 : Image d’iris-	- 18
I.3 : État du marché de la biométrie à l’échelle mondiale	- 21
II.1: Exemple d’histogramme d’une image	- 27
II.2: Exemple de normalisation d’histogramme	- 29
II.3: Exemple d’égaliseur d’histogramme	- 29-30
II.4: Exemple de filtre moyen	- 31
II.5: Exemple de filtre gaussien	
II.6: Exemple de filtre médian	- 33
II.7: Exemple du filtre de Nagao	
III.1: Comparaison entre différents systèmes biométriques	- 39
III.2: Architecture d’un système de reconnaissance de visage	- 40
IV.1: La concentration d’énergie par la DCT	- 48
IV.2: Séparabilité de la DCT 2-D	- 49
IV.3: DCT vs KLT en variance	- 50
IV.4: Application de la DCT par bloc	- 51
IV.5: Application de la DCT par bloc	- 53
IV.6: Reconnaissance par DCT	- 54
V.1: Exemple de projection suivant PCA	- 57
V.2: Processus de reconnaissance par PCA	- 62
V.3: Processus de reconnaissance par (2D)PCA	- 64
V.4: la PCA dans le domaine DCT	- 65
VI.1: Construction d’une image diagonale dans le cas où les lignes sont supérieures aux colonnes	- 69
VI.2: Construction d’une image diagonale dans le cas où les colonnes supérieures aux lignes	
VI.3: Construction d’une image diagonale	- 70
VI.4: Application du diagonal sur une image	
VI.5: Application de reconnaissance par diagonal PCA	- 72
VI.6: Processus de reconnaissance par diagonal (2D)PCA	- 73
VI.7: Processus de reconnaissance par diagonal PCA	- 75
VI.8: Processus de reconnaissance par diagonal (2D)PCA dans le domaine DCT-	- 77
VI.9: Processus de reconnaissance par diagonal PCA dans le domaine DCT-	- 78
VII.1: Diagramme de cas d’utilisation pour l’administrateur	- 83
VII.2: Diagramme de séquence pour la gestion de base de données	- 84
VII.3: Diagramme de séquence pour l’apprentissage	- 85
VII.4: Diagramme de séquence pour un test global	
VII.5: Diagramme de séquence pour un test d’un Person	- 86
VII.6: Diagramme de classe pour la gestion d’une base de données	- 87
VII.7: Diagramme de collaboration pour la gestion de BDD	- 93
VII.8: Diagramme de collaboration pour l’apprentissage	
VII.9: Diagramme de collaboration pour un test	- 94
VII.10: Diagramme de collaboration pour une personne test	
VII.11: Schéma conceptuel du système	- 97
VII.12: La forme principale du système	- 98
VII.13: La forme de base de données	- 99

VII.14: Forme de l'apprentissage	-	-	-	-	-	-	-	-	99
VII.15: Forme de test	-	-	-	-	-	-	-	-	100
VII.16: Forme d'acquisition	-	-	-	-	-	-	-	-	101
VIII.1: Variation en inclinaison, échelle et expressions faciales de la base ORL	-	-	-	-	-	-	-	-	104
VIII.2: Variation en coupe de cheveux, barbes, lunettes et luminance de la base ORL	-	-	-	-	-	-	-	-	105
VIII.3: Variation en inclinaison, échelle et expressions faciales de la base Blida									
VIII.4: Variation en coupe de cheveux, barbes, lunettes et luminance de la base Blida									
VIII.5: Application de luminance 60 sur une image	-	-	-	-	-	-	-	-	106
VIII.6: Taux de reconnaissance en fonction du nombre de vecteurs PCA	-	-	-	-	-	-	-	-	107
VIII.7: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	108
VIII.8: Taux de reconnaissance en fonction du nombre de vecteurs DiaPCA	-	-	-	-	-	-	-	-	109
VIII.9: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	110
VIII.10: Taux de reconnaissance en fonction du nombre de vecteurs (2D)PCA	-	-	-	-	-	-	-	-	111
VIII.11: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	112
VIII.12: Taux de reconnaissance en fonction du nombre de vecteurs Dia(2D)PCA	-	-	-	-	-	-	-	-	113
VIII.13: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	114
VIII.14: Taux de reconnaissance en fonction du chevauchement et du nombre de coefficients DCT	-	-	-	-	-	-	-	-	116
VIII.15: Taux de reconnaissance en fonction de la luminance et du nombre de coefficients DCT	-	-	-	-	-	-	-	-	117
VIII.16: Taux de reconnaissance en fonction de la luminance et du nombre de coefficients DCT	-	-	-	-	-	-	-	-	118
VIII.17: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DCTGlobal	-	-	-	-	-	-	-	-	119
VIII.18: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DiaDCTG	-	-	-	-	-	-	-	-	119
VIII.19: Taux de reconnaissance en fonction du chevauchement et du nombre de coefficients DCT	-	-	-	-	-	-	-	-	120
VIII.20: Taux de reconnaissance en fonction des vecteurs PCA et des coefficients DCTBloc	-	-	-	-	-	-	-	-	121
VIII.21: Zoom de la figure VII.20									
VIII.22: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	122
VIII.23: Taux de reconnaissance en fonction des vecteurs DiaPCA et des coefficients DCTBloc	-	-	-	-	-	-	-	-	123
VIII.24: Zoom de la figure VII.23									
VIII.25: Taux de reconnaissance en fonction de la luminance	-	-	-	-	-	-	-	-	124
VIII.26: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DCT	-	-	-	-	-	-	-	-	124
VIII.27: Taux de reconnaissance en fonction de la taille du bloc DCTG et de la luminance	-	-	-	-	-	-	-	-	125

INTRODUCTION

Le besoin des sociétés en sécurité va en augmentant aussi bien sur le web qu'au sein des entreprises : mots de passe, badges, cartes à puce ne cessent de se multiplier. Mais, paradoxalement, ces moyens de sécurité s'avérant insuffisants, favorisent la multiplication des moyens de fraude. Ce qui conduit, aujourd'hui, les entreprises à songer à des moyens considérés anciennement comme relevant de science-fiction.

Ainsi, on assiste aujourd'hui, grâce au développement technologique, au recours à un moyen d'identification des personnes qui s'est avéré plus rapide et moins enclin à la fraude et donc plus fiable, connu sous le nom de la *biométrie*. Les anciens moyens de sécurité, parmi lesquels nous citerons les mots de passe, ont des problèmes de mémorisation que la biométrie, par exemple, ne connaît pas.

Aussi, le CDTA (Centre de Développement des Technologies Avancées) ayant enregistré une forte demande des entreprises algériennes en sécurité, s'est lancé dans la reconnaissance biométrique. Son projet étant de fabriquer des cartes magnétiques, le CDTA, en collaboration avec les universités, a formé des équipes de recherche qui travaillent sur des sous-projets : reconnaissance vocale ou locuteur, reconnaissance de visage et reconnaissance de l'empreinte digitale.

Le projet de la reconnaissance de visage nous a motivés dans la mesure où il intéresse des entreprises où nous souhaitons apporter notre contribution plus tard.

Une équipe du CDTA en quête d'algorithmes rapides, simples et fiables nous a dirigés vers une étude comparative entre deux méthodes de reconnaissance de visage, la DCT-PCA et la DCT-DiagPCA. La DCT (Transformée en Cosinus Discrète, en anglais Discrete Cosines Transform) est une méthode de compression d'images, c'est une méthode très courante, d'ailleurs elle est à l'origine du JPEG et du MPEG (voir définitions dans notes de bas de pages Chap. IV, pp.41-42) ; la PCA (Analyse en Composantes Principales, en anglais Principal Component Analysis) est une approche statistique qui a fait ses preuves dans la reconnaissance des visages ; la DiagPCA est

une nouvelle version de la PCA, mise au point afin de compléter la première version. Quant à la DCT-PCA, c'est une combinaison de la méthode de compression et de l'approche statistique qui a été mise au point afin d'accélérer le processus de la reconnaissance des visages car la PCA requiert beaucoup de temps.

La DiagPCA ayant donné de meilleurs résultats que la PCA, nous sommes partis de l'hypothèse suivante : si l'on combinait la DCT à la DiagPCA donnerait-elle un meilleur taux de reconnaissance que la DCT-PCA ?

Et pour vérifier notre hypothèse, nous avons suivi la démarche suivante :

D'abord, nous avons comparé les deux types de DCT (DCT Bloc et DCT Globale) et ce, afin de voir les caractéristiques de chacune et quel(s) avantage(s) et/ou inconvénient(s) présente chacune d'elles, ensuite nous avons comparé la PCA et la DiagPCA, la (2D)PCA et la Diag(2D)PCA, le concept du 2D étant relatif à la forme matricielle des images et non pas vectorielle comme c'est le cas pour la PCA classique, et enfin nous avons comparé les différentes combinaisons de la DCT et de la PCA jusqu'à arriver à la combinaison DCT Globale-Diag(2D)PCA.

Outils utilisés :

- Microsoft Visual Studio.Net
- SQL Server 2000
- Matlab

Bases de données visage utilisées :

- ORL (Olivetti Research Laboratory) ;
- Blida Visage : Etudiants de l'université Saad Dahlab de Blida

Mots-clés : Biométrie, Reconnaissance de visage, Luminance, Chevauchement, Taux de reconnaissance.

PARTIE 1 :
Biométrie et reconnaissance de
visage

Chapitre I :
Introduction à la Biométrie

Introduction :

Le terme de *Biométrie* est utilisé dans le sens d'*étude quantitative des êtres vivants*. Il désigne le plus souvent l'*identification des personnes* en fonction de caractéristiques biologiques telles que les empreintes digitales, les traits du visage, etc.

[R1]

Depuis quelques années, la biométrie a connu un essor considérable dans le domaine de la sécurité, boosté (propulsé) par les media, les avancées technologiques, et la forte demande d'une sécurité fiable à tous les niveaux même à titre personnel.

Pourquoi utiliser la Biométrie ?

Traditionnellement, pour vérifier l'identité, on utilise des codes, des mots de passe, des badges, des cartes à puce et autres qui présentent des inconvénients :

- Problème de mémorisation.
- Risque d'être utilisés par des personnes non autorisées ...etc.

L'avantage de l'identification biométrique est que chaque individu a ses propres caractéristiques physiques qui ne peuvent être changées, perdues ou volées. Ainsi, l'un des buts de la biométrie est de protéger les consommateurs contre la fraude ou le vol.

I.2. Méthode d'authentification biométrique :

La multitude des caractères biométriques de l'être humain a donné naissance à plusieurs systèmes d'authentification, chacun reposant sur un caractère morphologique ou comportemental. La méthode d'authentification biométrique peut être utilisée en complément ou en remplacement des techniques classiques.

Généralement, on distingue deux catégories de méthodes d'authentification biométrique :

1. Méthodes basées sur les *caractéristiques comportementales* : dynamique de la signature (ex vitesse de déplacement du stylo, accélérations, pression exercée, inclinaison, ...etc.), façon d'utiliser un clavier d'ordinateur (pression exercée, vitesse de frappe), manière de marcher...etc.
2. Méthodes basées sur les *caractéristiques morphologiques* : empreintes digitales, formes de la main, traits du visage, dessin du réseau veineux de l'œil...etc. Ces éléments morphologiques ont l'avantage d'être stables dans la vie d'un individu

et ne subissent pas autant les effets du stress, que l'on retrouve dans l'authentification comportementale.

Parmi ces systèmes, certains ont prouvé leur fiabilité et leurs performances, d'autres sont toujours en cours d'évolution.

I.3. Présentation des différents systèmes biométriques :

I.3.1. Systèmes morphologiques :

I.3.1.1. Empreintes digitales :

Il s'agit d'une des premières biométries utilisées dans des machines d'authentification, c'est la technologie la plus mature pour la reconnaissance des criminels.

La formation des empreintes dépend des conditions initiales du développement embryogénique, ce qui les rend uniques à chaque personne et même à chaque doigt.

L'image d'empreinte est prise soit traditionnellement par un scanning du doigt couvert d'encre soit en utilisant un dispositif d'acquisition d'image spécifique.

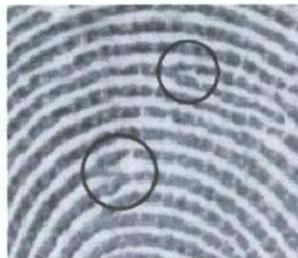


Figure I.1 : Image d'empreinte

Les images d'empreintes digitales sont relativement faciles à traiter, car la plupart de l'information est contenue dans la forme des lignes. Le traitement est généralement effectué sur des images binaires. Autrefois utilisée par les forces de sécurité, cette technique se développe à présent dans le secteur grand public grâce à l'apparition de capteurs bon marché.

I.3.1.2. Géométrie de la main/du doigt :

Ce type de mesure biométrique est l'un des plus répandus aux États-Unis. Il consiste à mesurer plusieurs caractéristiques de la main (jusqu'à 90) telles que la forme de la main, la longueur et la largeur des doigts, les formes des articulations, les longueurs inter articulations, etc. La technologie associée à cela est principalement de l'imagerie infrarouge; d'une façon générale, le système présente des FAR (False Acceptation Rate, voir plus bas) assez élevés, surtout entre des personnes de la même famille ou bien encore des jumeaux.

I.3.1.3. Iris :

Si la couleur, la forme et l'apparence générale de l'iris est déterminée génétiquement, sa texture détaillée est propre à chaque individu, voire même à chaque œil. De plus, cette texture est stable et ne peut être modifiée sans perte importante des capacités visuelles. Une technique de description de cette texture basée sur des filtres de Gabor a été développée et a conduit à des applications commerciales (la compagnie qui détient les brevets sur la reconnaissance par l'iris s'appelle IriScan).

Cette technique est sûre et moins contraignante pour l'utilisateur que celle de la rétine, mais elle nécessite jusqu'à présent que l'utilisateur place son œil en bonne position par rapport à la caméra, ce qui a fortement limité son succès en Europe.

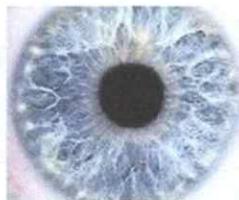


Figure I.2 : Image iris

Cependant, l'incorporation de techniques de localisation de l'œil permet de relaxer la contrainte à laquelle l'utilisateur est soumis et son utilisation est envisagée dans les distributeurs de billets de banque et pour l'accès sécurisé à Internet.

I.3.1.4. Rétine :

Il a été montré que chaque œil possède en sa rétine un arrangement unique des vaisseaux sanguins. La technique basée sur la rétine utilise la texture de ces vaisseaux. Elle est relativement ancienne et a été utilisée essentiellement dans des environnements de haute sécurité, comme l'accès aux sites nucléaires militaires.

Cette méthode requiert une collaboration étroite de la part du sujet, car il doit placer son œil extrêmement près de la caméra (quelques centimètres). Le recours à cette méthode se limite aux établissements les plus hautement sécurisés (exemple, le Ministère de la Défense).

I.3.1.5. Visage :

Il s'agit de capter la forme du visage d'un individu et d'en extraire certaines informations jugées évidentes pour l'authentification.

Selon le système utilisé, l'individu doit être positionné devant l'appareil ou peut être en mouvement à une certaine distance. Les données biométriques qui sont obtenues sont par la suite comparées au fichier référence.

Au début des années 1970, la reconnaissance par le visage était principalement basée sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton, la forme,...etc. Depuis les années 1990, les différentes technologies utilisées exploitent toutes les découvertes effectuées dans le domaine du traitement d'image et de l'analyse de données.

I.3.1.6. Reconnaissance vocale :

La reconnaissance d'un locuteur offre l'avantage d'être bien acceptée par l'utilisateur, quelle que soit sa culture. De plus, s'il s'agit de sécuriser une transaction téléphonique, la voix est la seule information disponible.

On distingue les systèmes à texte prédéterminé (texte dépendant) où l'utilisateur doit répéter un texte qu'il ne choisit pas et les systèmes où la personne peut parler librement (texte indépendant).

La performance de cette méthode est sujette à la qualité du signal, qui dépend de la variabilité de la voix du locuteur dans le temps comme dans le cas de maladie (rhume), d'états émotionnels (angoisse, joie) et de l'âge ; des conditions d'acquisition de la voix telles que le bruit et la réverbération ; de la qualité des équipements tels que le

microphone, sans oublier le fait que différentes personnes peuvent avoir des voix similaires.

I.3.2. Systèmes comportementaux :

I.3.2.1. Dynamique des signatures :

Ce type de biométrie est à l'heure actuelle peu utilisé, mais ses défenseurs espèrent l'imposer assez rapidement pour des applications spécifiques (documents électroniques, rapports, contrats...).

Le procédé est habituellement combiné à une palette graphique (ou équivalent) munie d'un stylo.

Ce dispositif va mesurer plusieurs caractéristiques lors de la signature, telles que la vitesse, l'ordre des frappes, la pression, les accélérations, le temps total et tout ce qui peut permettre d'identifier une personne de la façon la plus sûre possible, vu que les données de la signature sont multi changeantes.

I.3.2.2. Dynamique de la frappe au clavier :

La dynamique de la frappe au clavier est une caractéristique propre de l'individu, c'est en quelque sorte la transposition de la graphologie aux moyens électroniques.

C'est une méthode qui utilise un moyen non intrusif qui exploite un geste naturel, mais elle dépend de l'état physique de la personne (âge, maladies,...).

Les paramètres pris en compte sont généralement la vitesse de frappe, la mesure des temps de frappe et la pause entre chaque mot.

I.4. État du marché de la biométrie :

Selon le cabinet américain Frost & Sullivan, se basant sur les sources de l'International Biometric Group, les revenus globaux de la biométrie ont une croissance estimée de 2.1 milliards de dollars en 2006 à 5.7 milliards en 2010.

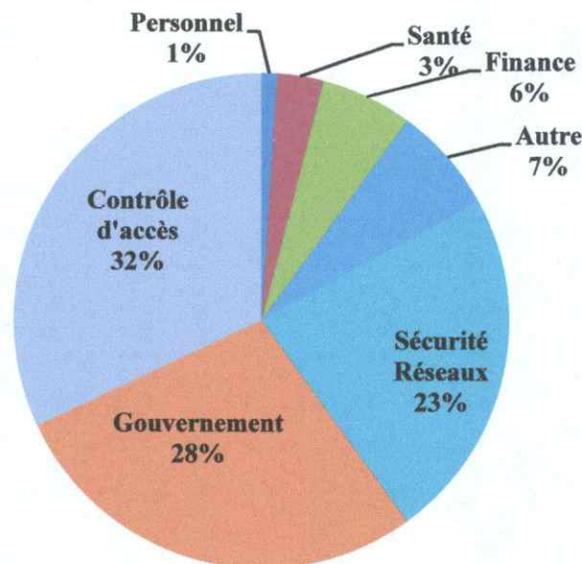


Figure 1.3 : État du marché de la biométrie à l'échelle mondiale

Ce marché autrefois dominé par les besoins gouvernementaux est désormais supplanté par les besoins des secteurs civils. Toujours selon cette même source, en termes de typologie biométrique, l'empreinte digitale représente 43.6% du marché en 2006 et y occupe une place dominante. Elle est suivie par l'identification de visage dont la part s'élève à 19%.

Les projections relatives à l'identification par l'iris prévoient un chiffre d'affaires potentiel mondial de 250 millions de dollars d'ici 2008.

L'Asie et l'Amérique seraient les zones géographiques porteuses des plus grands marchés en termes de produits biométriques et services associés. On constate l'émergence des systèmes multimodaux qui, d'ici 2010, devraient représenter approximativement 5% du marché de la biométrie.

Conclusion :

La biométrie est une véritable alternative aux mots de passe et autres identifiants. Elle permet de vérifier que l'utilisateur est bien la personne qu'il prétend être. Cette technologie est en pleine croissance et tend à s'associer à d'autres technologies comme la carte à puce.

Après avoir défini ce qu'est la biométrie, nous allons présenter dans le chapitre suivant quelques notions de base sur le traitement d'images.

Chapitre II :
Notions de base sur le
traitement d'image

II.1. Définition générale :

Le traitement d'images désigne en informatique l'ensemble des traitements automatisés qui permettent, à partir d'images numérisées, de produire d'autres images numériques ou d'en extraire de l'information. Il s'agit donc d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo, tout en opérant dans le domaine numérique.

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant :

1. la résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions,
2. les réglages optiques utilisés, (dont la mise au point) qui déterminent par exemple la netteté de l'image,
3. les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées,
4. le bruit de la chaîne de transmission d'image.

II.2. Définition d'une image numérique :

On désigne sous le terme d'image numérique toute image (dessin, icône, photographie ...) *acquise, créée, traitée* ou *stockée* sous forme binaire (suite de 0 et de 1) [R2]:

- Acquise par des dispositifs comme les scanners, les appareils photo ou caméscopes numériques, les cartes d'acquisition vidéo (qui numérisent directement une source comme la télévision).
- Créée directement par des programmes informatiques, via la souris, les tablettes graphiques ou par la modélisation 3D (ce que l'on appelle par abus de langage les « images de synthèse »).
- Traitée grâce à des outils informatiques. Il est facile de la modifier en taille, en couleur, d'ajouter ou supprimer des éléments, d'appliquer des filtres variés, etc.

- Stockée sur un support informatique (disquette, disque dur, CD-ROM, ...)

II.3. Caractéristiques d'une image numérique :

L'image numérique est constituée d'une matrice de points, appelés pixels (contraction de l'anglais Picture element) bien adaptée à l'affichage sur écran informatique. Une image se définit par :

II.3.1. Pixel :

Le pixel représente le plus petit point distinguable dans une image. Chaque pixel possédant une teinte, c'est la juxtaposition (*corrélation implicite*) des différents pixels qui produit une image.

II.3.2. Résolution :

Celle-ci définit la clarté ou finesse de détail atteinte par un moniteur ou une imprimante dans la production d'images, elle est exprimée par un nombre de pixels par unité de longueur de la structure à numériser [R3].

II.3.3. Dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

II.3.4. Représentation des couleurs : [R4]

Il existe plusieurs modes de codage informatique des couleurs, le plus utilisé pour le maniement des images est l'espace colorimétrique Rouge, Vert, Bleu (RVB ou RGB¹). Cet espace est basé sur une synthèse additive des couleurs, c'est-à-dire que le mélange des trois composantes R, V, et B. On peut citer aussi d'autres modes tels que :

- Cyan, Magenta, Jaune, Noir (CMJN ou Cyan, Magenta, Yellow, Key (Black) CMYK), couleurs primaires utilisées principalement pour l'impression et basées sur une synthèse soustractive des couleurs ;
- Teinte, Saturation, Luminance (TSL ou HSL Hue Saturation Lightness), où la couleur est codée suivant le cercle des couleurs.
- base de couleur optimale YUV, Y représentant la luminance, U et V deux chrominances orthogonales.

II. 3.5. Image en niveaux de gris :

Les images réalisées en niveaux de gris utilisent les teintes grises allant du blanc au noir. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la couleur de chaque pixel de l'image. Par exemple avec 8 bits, on aura 256 niveaux de gris compris entre 0 et 255, 0 représente le blanc et 255 le noir. Le Gris = $0.299 * \text{Rouge} + 0.587 * \text{Vert} + 0.114 * \text{Bleu}$ (norme C.I.E (Commission Internationale de l'Éclairage) pour les couleurs non-linéaires).

II.3.6. Histogramme d'une image : [R5]

L'histogramme représente la répartition des pixels en fonction de leur niveau de gris. Soit une matrice d'image A , Le rôle de l'histogramme est de donner le nombre de pixels en A qui ont la valeur L tel que $L= 0, \dots, 255$.

¹ RGB : Red, Green, Blue

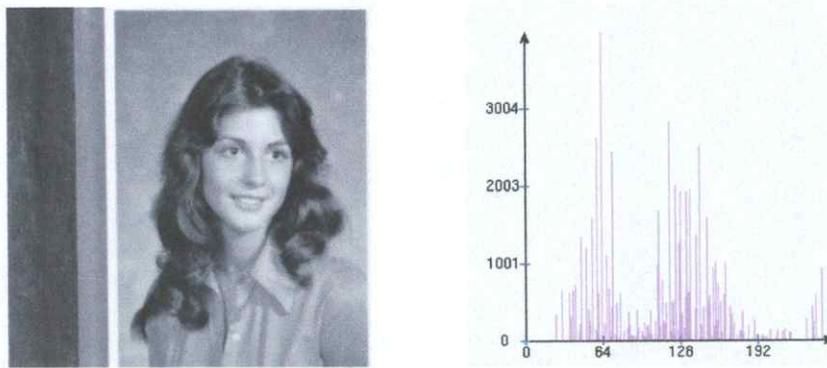


Figure II.1 : Exemple d'histogramme d'une image [MANZ]

II.3.7. Contours et textures :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative [R6]. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes [R7].

II.3.8. Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

II.3.9. Contraste :

C'est l'opposition entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si $L1$ et $L2$ sont les degrés de luminosité respectivement de deux zones voisines $A1$ et $A2$ d'une image, le contraste C est défini par le rapport :

$$C = \frac{L1 - L2}{L1 + L2} \quad (1)$$

II.3.10. Convolution :

C'est l'opérateur de base du traitement linéaire des images. Soit I une image numérique.

Soit h une fonction de $(x_1, x_2) \times (y_1, y_2)$ à valeurs réelles.

La *convolution* de I par h est définie par :

$$(I * h)(x, y) = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h(i, j) * I(x - i, y - j) \quad (2)$$

La fonction h est dite noyau de la convolution, les nouvelles valeurs du pixel sont calculées par *produit scalaire* entre le noyau de convolution et le *voisinage* correspondant du pixel.

II.4. Prétraitement :II.4.1. Traitement à base d'histogramme : [R5]

Nous présentons dans la suite quelques traitements d'analyses effectuées uniquement à partir de l'histogramme. Retenons que certains de ces traitements sont souvent calculés au niveau des capteurs et qu'en général leur pertinence est très intimement liée aux conditions d'acquisition.

I.4.1.1. Normalisation :

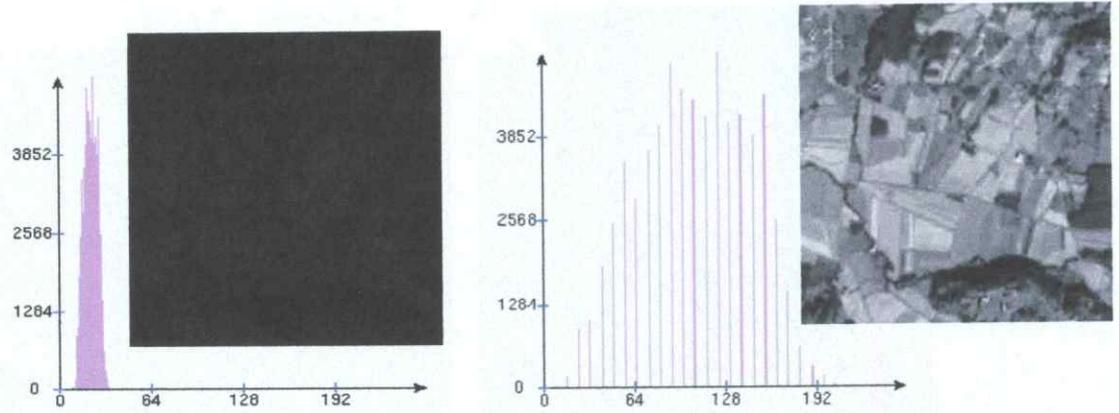
La normalisation d'histogramme, ou expansion de dynamique, est une transformation affine du niveau de gris des pixels de telle sorte que l'image utilise toute la dynamique de représentation.

$$f_{new}(x, y) = (f(x, y) - N_{min}) \cdot \frac{2^D - 1}{N_{max} - N_{min}}$$

D : Dynamique (Nbre de Bits)

N_{max} : la plus grande valeur des pixels de l'image

N_{min} : la plus petite valeur des pixels de l'image



Histogramme d'origine

Histogramme normalisé

Figure II.2 : Exemple de normalisation d'histogramme [R5]

II.4.1.2. Egalisation :

L'égalisation d'histogramme est une transformation des niveaux de gris dont le principe est d'équilibrer le mieux possible la distribution des pixels dans la dynamique (idéalement, on cherche à obtenir un histogramme plat).

La technique classique consiste à rendre « le plus linéaire possible » l'histogramme cumulé de l'image en utilisant la transformation suivante :

$$f_{new}(x, y) = (2^D - 1) \cdot \frac{HC(f(x, y))}{wh} \quad (3)$$

D : Dynamique (Nbre de Bits)

(w,h) : Dimension de l'image

HC(.) : Histogramme cumulé

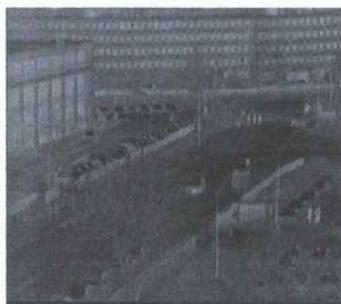
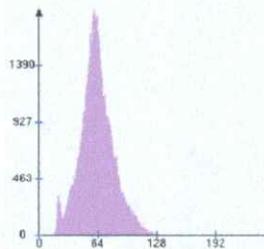
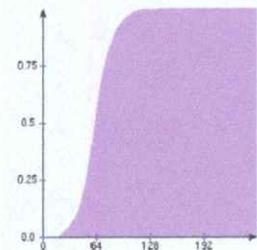


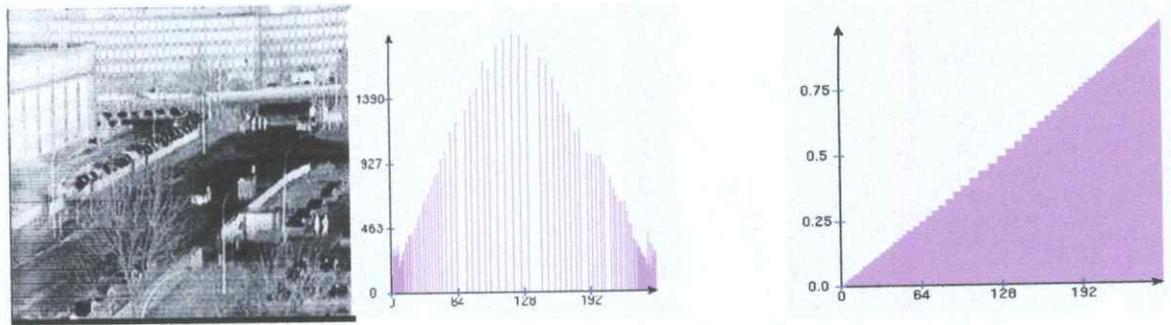
Image originale f(x,y)



Histogr de f



Histogr cumulé de f



Après Egalisation f_{new}

Histogr de f_{new}

Histogr cumulé de f_{new}

Figure II.3 : Exemple d'égaliseur d'histogramme

II.4.2. Filtrage :

II.4.2.1. Filtres linéaires :

C'est le résultat d'une combinaison linéaire des plus proches voisins d'un pixel, parmi les filtres linéaires existant, nous pouvons citer :

- Filtre moyen [R8] :

C'est un filtre qui, comme son nom l'indique, fait la moyenne entre toutes les valeurs de pixels avoisinant un point. Il permet de lisser l'image, réduit le bruit, réduit les détails non importants et brouille ou rend floue l'image.

Exemple : Filtre moyen, de taille 3 :

$$1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Son intérêt est qu'il ne change pas trop le contour. On peut mentionner aussi que plus le filtre est grand, plus le lissage devient important et plus le flou s'accroît, ce qui pourrait engendrer des conséquences néfastes sur la qualité de l'image.

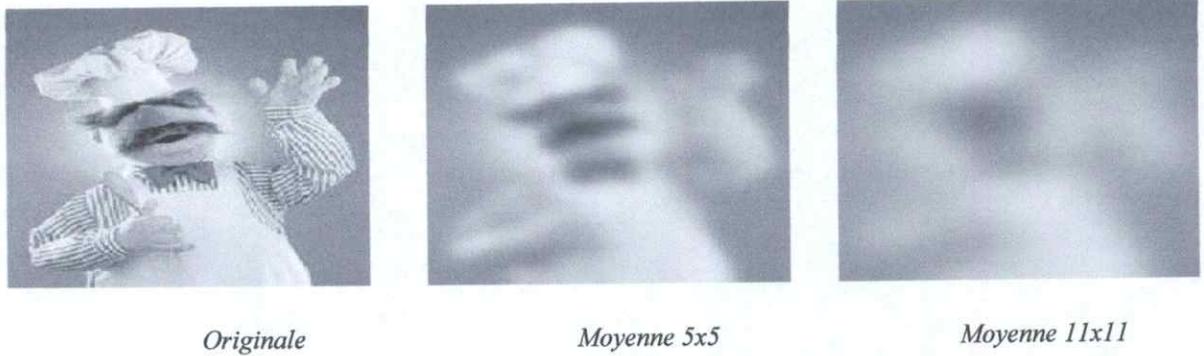


Figure II.4 : Exemple de filtre moyenne [R8]

- Filtre gaussien [R5]:

C'est un filtre qui s'appuie sur la version échantillonnée normalisée de la fonction gaussienne

$$h(x, y) = \frac{1}{2\pi\sigma^2} \times e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Le filtre gaussien donnera un meilleur lissage et une meilleure réduction du bruit que le filtre moyen. Exemple de filtre gaussien de taille 5 :

$$\frac{1}{1444} \times \begin{pmatrix} 1 & 9 & 18 & 9 & 1 \\ 9 & 81 & 162 & 81 & 9 \\ 18 & 162 & 324 & 162 & 18 \\ 9 & 81 & 162 & 81 & 9 \\ 1 & 9 & 18 & 9 & 1 \end{pmatrix}$$

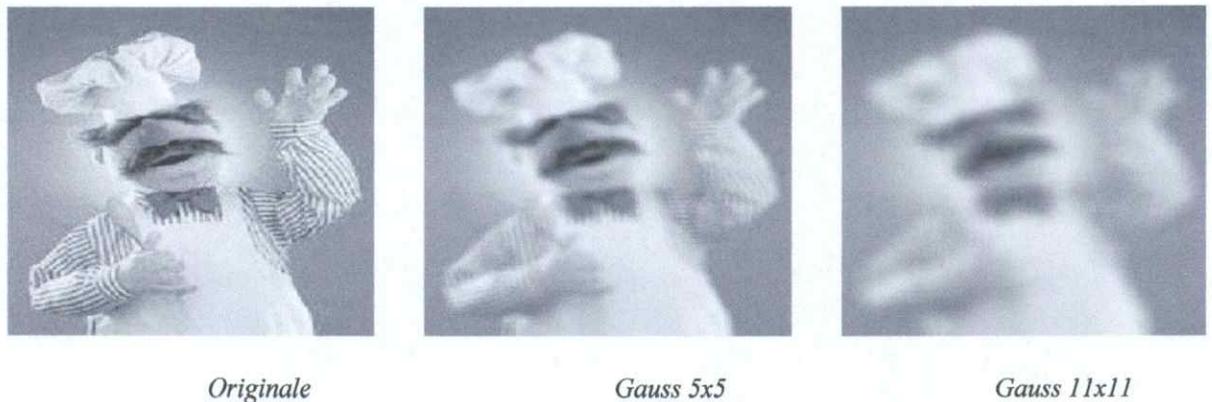


Figure II.5 : Exemple de filtre gaussien [R8]

II.4.2.2. Filtres non linéaires : [R5]

Contrairement aux filtres linéaires, les filtres non linéaires ne sont pas le résultat d'une combinaison linéaire de leurs voisins ainsi ils ne peuvent pas s'implémenter comme un produit de convolution. Deux aspects du lissage sont concernés par le filtrage non linéaire :

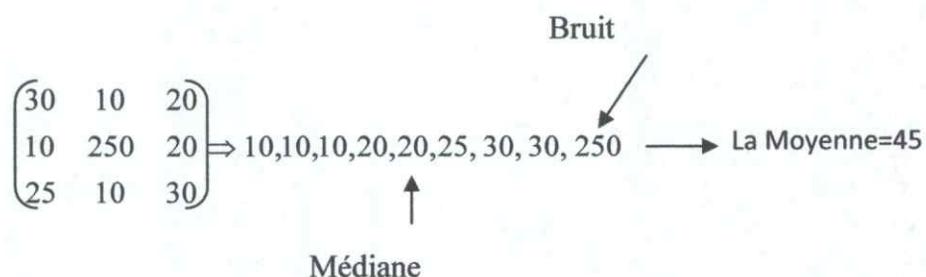
1. Le bruit impulsionnel : les filtres linéaires éliminent mal les valeurs aberrantes.
2. L'intégrité des frontières : on souhaiterait éliminer le bruit sans rendre floues les frontières des objets.

Nous pouvons citer et définir parmi les filtres linéaires les plus utilisés :

- Filtre médian :

Il permet d'éliminer certains types de bruits (poivre et sel), son principe est de remplacer la valeur d'un pixel par la valeur médiane de la suite mathématique constituée des valeurs des pixels avoisinants à ce point. Pour une meilleure performance de ce filtre, on commence par trier les valeurs des pixels du voisinage, suivra ensuite la détermination de la médiane et enfin l'affectation de cette valeur au pixel. La principale fonction du filtre médian est de forcer des points avec des intensités très distinctes pour être comme leurs voisins, ainsi que d'éliminer réellement les intensités transitoires qui apparaissent isolées dans la zone de masque.

Exemple de filtre médian de taille 3 :



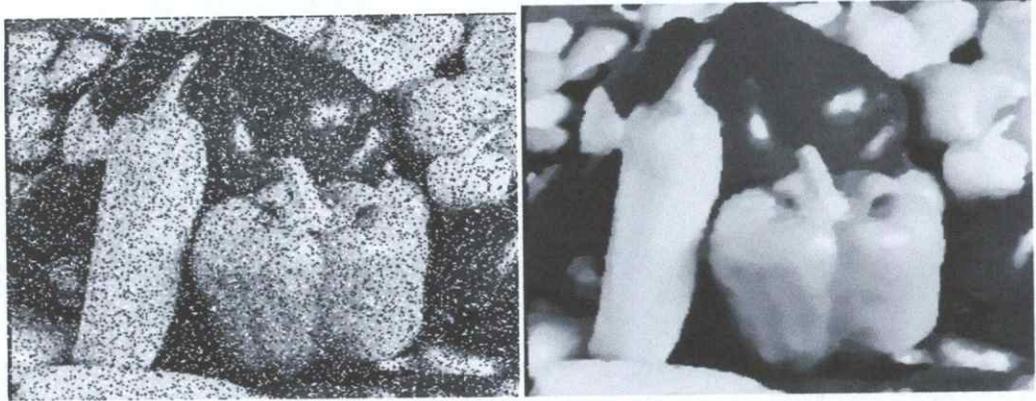


Figure II.6 : Exemple de filtre médian [R5]

- Filtre de Nagao : [R9]

Il s'agit d'un filtre utilisé fréquemment dans les images comportant de très fortes structures artificielles (images aériennes de zones urbaines, vision industrielle, etc). Il découpe une fenêtre 5x5 centrée sur le pixel en 9 fenêtres de 3x3 chacune et mesure sur chacune de ces fenêtres une valeur de l'*homogénéité* (variance par exemple). Le pixel central est alors remplacé par la valeur de la zone la plus homogène.

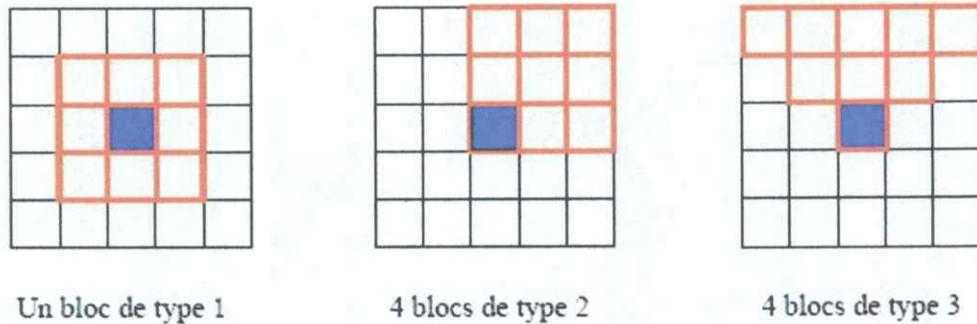


Figure II.7 : Exemple du filtre de Nagao [R9]

II.4.2.3. Filtres de détection de contours [R10] :

La détection de contour est une étape préliminaire pour de nombreuses applications de l'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image.

Les contours dans une image proviennent des :

1. Discontinuités de la fonction de réflectance (texture, ombre)
2. Discontinuités de profondeur (bords de l'objet)

Il existe plusieurs méthodes de détection de contours, nous citerons à cet effet trois classes, suivant la manière d'estimer les dérivées de la fonction d'intensité :

- Différences finies :

Une image est discrète par nature. Les premières approches consistent à approximer les dérivées par différence, ces dérivées sont calculées par convolution de l'image avec un masque de différences. Nous citerons à cet effet les filtres de *Roberts*, *Prewitt*, *Sobel*, *Kirsh* et *Robinson*.

- Filtrage optimal :

Les dérivations présentées consistent à convoluer l'image par des masques de petites dimensions. Ces approches sont donc dépendantes de la taille des objets traités, elles sont aussi très sensibles au bruit. Un autre type d'approche plus récente repose sur la définition de critères d'optimalité de la détection de contours; ces critères débouchant sur des filtres de lissage *optimaux*. Nous citerons parmi les filtres correspondants, *Canny*, *Shen-Castan*, *Deriche* et *Marr*.

- Modélisation de la fonction d'intensité :

Les différents filtres cités ci-dessus permettent de calculer le gradient ou le laplacien d'une image mais ne donnent pas des points de contours. Un traitement ultérieur est nécessaire, ce traitement étant dépendant du type d'approche choisi, approche par Gradient ou approche par le Laplacien [R10].

Conclusion :

A travers ce chapitre, nous avons défini des notions courantes dans le domaine du traitement d'images, à savoir les caractéristiques des images numériques, les prétraitements souvent utilisés dans le domaine de l'imagerie et les notions de filtrage. Nous allons présenter dans le chapitre qui suit l'état de l'art des différents systèmes de reconnaissance de visage.

PARTIE 2 :

**Etat de l'art sur les systèmes
de reconnaissance**

Chapitre III :
Etat de l'art sur les systèmes
de reconnaissance

Introduction :

La biométrie est une technique globale visant à établir l'identité d'une personne en mesurant une de ses caractéristiques physiques. Il peut y avoir plusieurs types de caractéristiques physiques, les unes plus fiables que d'autres, mais toutes doivent être infalsifiables et uniques pour pouvoir être représentatives d'un et un seul individu.

Il existe plusieurs possibilités pour prouver l'identité d'une personne :

1. ce qu'il possède (carte, badge, document) ;
2. ce qu'il sait (un nom, un mot de passe) ;
3. ce qu'il est (empreintes digitales, main, visage...).

Il s'agit de la biométrie.

Les deux premiers moyens d'identification peuvent être utilisés pour usurper l'identité d'un tiers.

La biométrie permet l'identification ou l'authentification à partir de données reconnaissables et vérifiables, propres à chaque individu. Ces caractéristiques collectées doivent être :

- universelles (exister chez tous les individus),
- uniques (permettre de différencier un individu par rapport à un autre),
- permanentes (autoriser l'évolution dans le temps),
- enregistrables (collecter les caractéristiques d'un individu avec l'accord de celui-ci),
- mesurables (autoriser une comparaison future).

III.1. Système de reconnaissance biométrique: [R11]

Un système biométrique est un système d'identification ou de vérification d'individus :

1. *Identification* : pour ce mode, on pose la simple question : « *qui suis-je ?* ». A partir de l'échantillon biométrique fourni, le dispositif recherche le gabarit correspondant dans sa base de données.
2. *Authentification* : dans ce mode, on pose la question : « *suis-je bien M. X ?* ». Techniquement, le dispositif vérifie par rapport à un code (identifiant) saisi sur un clavier, ou lu par le passage d'un badge (carte à puce, magnétique, proximité, etc.) que l'échantillon biométrique fourni correspond bien au gabarit désigné par l'identifiant.

III.2. Pourquoi le visage ?

La reconnaissance de visage 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.

Par rapport aux autres méthodes, la reconnaissance de 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).

Même si l'on peut penser que la reconnaissance de visage est une biométrie relativement peu sûre, car le signal acquis est sujet à des variations beaucoup plus élevées que d'autres caractéristiques (variation de l'éclairage, changement de la position du visage, présence ou absence de lunettes et autres), ces dernières années ont connu le développement des techniques de traitements d'images telles que la détection du visage, la normalisation de l'éclairage, etc., sans oublier le développement des technologies des caméras numériques, ce qui réduit considérablement l'effet de ces risques.

Occupant une deuxième place dans l'industrie de la biométrie avec une part de 19% du marché après l'empreinte digitale, la reconnaissance de visage s'avère le bon compromis entre le coût et la précision.

Le graphe suivant montre une comparaison des techniques biométriques les plus utilisées en fonction des coûts et de la précision :

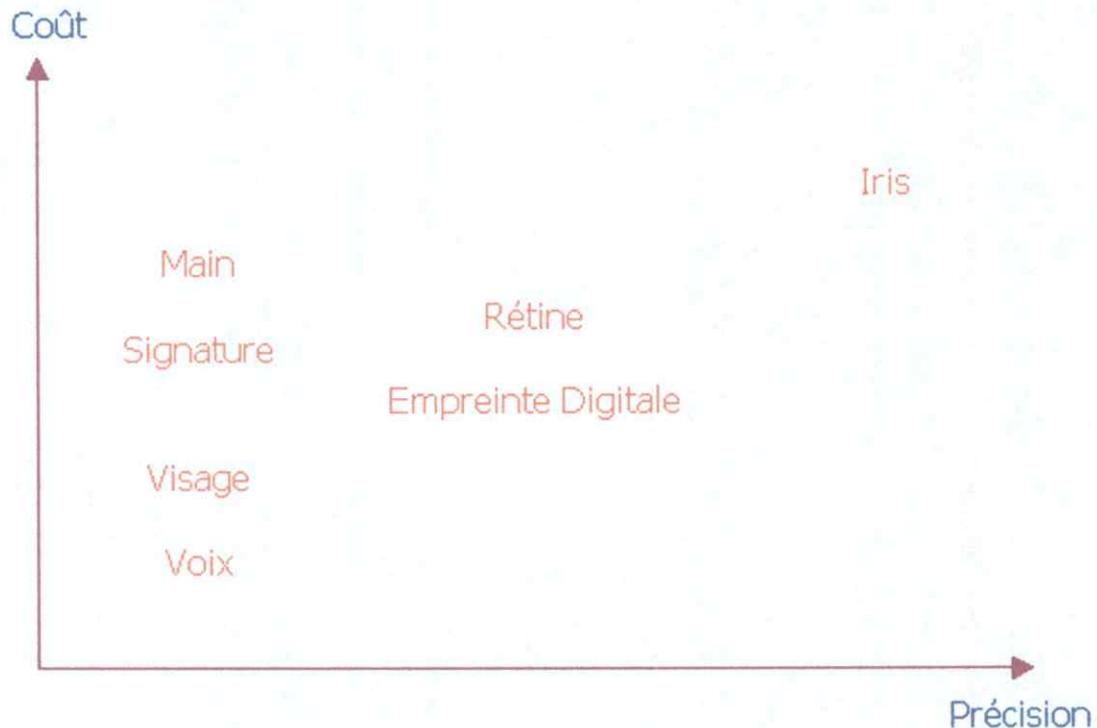


Figure III.1 : Comparaison entre différents systèmes biométriques

III.3. Processus de reconnaissance de visage :

Les systèmes de reconnaissance de visage reposent sur des algorithmes statistiques qui créent un modèle de chaque image visage. Les modèles ainsi obtenus sont regroupés pour former une base de modèles appelée modèle-apprentissage. L'étape de test consiste à ramener l'image-test dans le même domaine pour pouvoir comparer le modèle-test et les modèles-apprentissage sauvegardés au préalable.

Chaque système de reconnaissance de visage suit les étapes suivantes :

- Acquisition et Détection de visage
- Prétraitement
- Apprentissage

- Test
- Décision (dans le cas de vérification)

Le schéma suivant montre les différentes étapes d'un système de reconnaissance de visage :

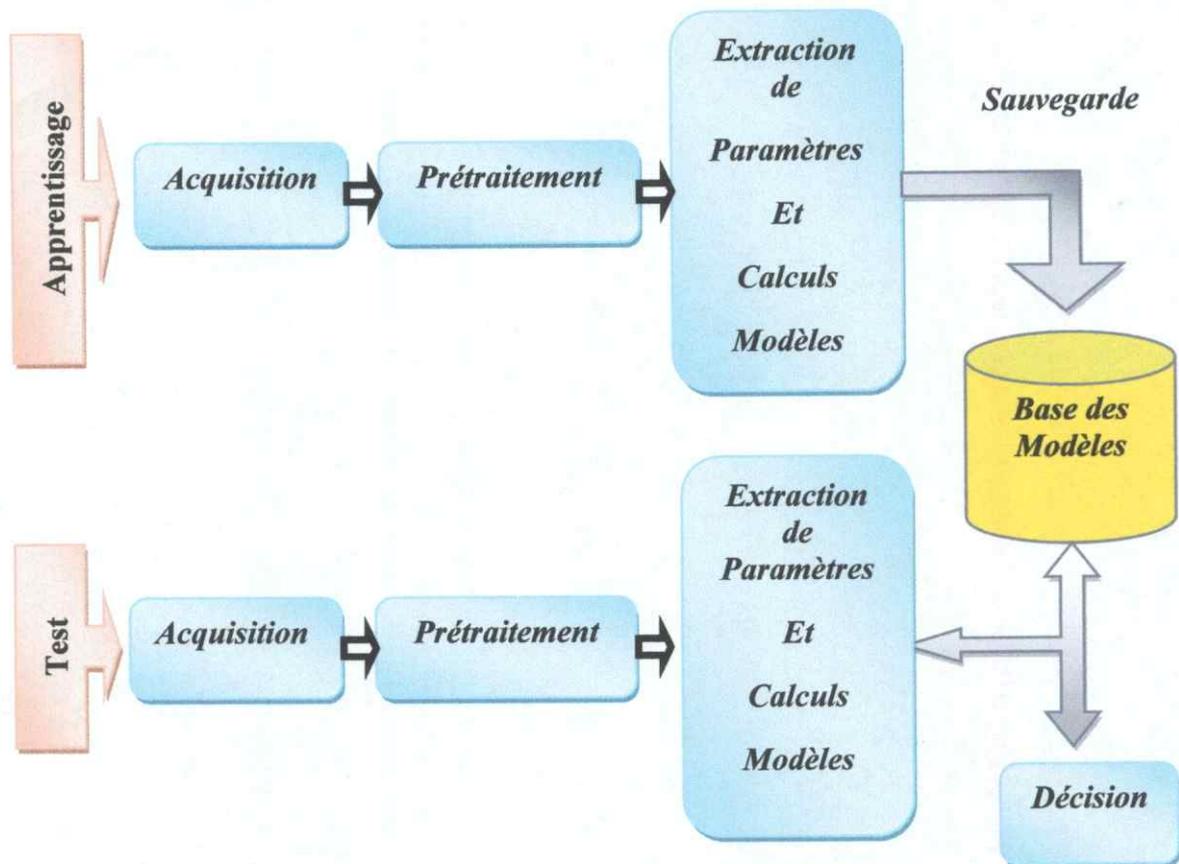


Figure III.2 : Architecture d'un système de reconnaissance de visage.

III.3.1. Acquisition :

Cette étape consiste à extraire l'image de l'utilisateur du monde extérieur dans un état statique à l'aide d'un appareil photo, ou dynamique à l'aide d'une caméra. Dans

le cas d'une camera, le système détecte le visage de l'utilisateur à travers un cadre coloré qui suit la couleur de la peau.

III.3.2. Prétraitement :

Le rôle de cette étape est d'éliminer les bruits causés par la qualité des dispositifs optiques ou électroniques lors de l'acquisition de l'image en entrée, dans le but de ne conserver que les informations essentielles. Il existe plusieurs types de prétraitement dans le système, nous n'utiliserons que trois :

- Normalisation
- Egalisation
- Luminance

Ces notions ont été définies dans le chapitre précédent.

III.3.3. L'extraction de paramètres :

L'étape d'extraction des paramètres représente le cœur du système de reconnaissance, elle consiste à effectuer le traitement de l'image dans un autre espace de travail plus simple qui assure une meilleure exploitation de données. Cette étape permet l'utilisation, seulement, des informations utiles, discriminantes et non redondantes.

Généralement, le changement d'espace se fait à l'aide des transformations orthogonales qui assurent une bonne et unique représentation de l'image dans le nouvel espace.

Il existe plusieurs transformations dont nous citerons :

III.3.3.1. Transformation en Cosinus Discrète (DCT) :

La DCT fait partie de ces transformations, c'est une technique très populaire dans le domaine du traitement de signal et de l'image, elle a été appliquée la première fois dans la compression d'images en 1974 par Ahmed *et al*, transformant la représentation spatiale du signal en une représentation fréquentielle.

La DCT possède en effet une excellente propriété de "regroupement" de l'énergie : l'information est essentiellement portée par les coefficients de basses fréquences, et pour les images naturelles, la DCT est la transformation qui se rapproche le plus de la transformée de Karhunen-Loève (*KLT*) qui fournit une décorrélation optimale des coefficients pour un signal donné.

Nous détaillerons cette approche dans le chapitre suivant.

III.3.3.2. Analyse en Composante Principale (PCA ou Eigen Faces) :

L'algorithme PCA est né des travaux de Turk et de Pentland, en 1991 [R12]. Il est aussi connu sous le nom d'Eigenfaces (visages propres) car il utilise des vecteurs propres et des valeurs propres désignés respectivement en anglais, Eigenvectors et Eigenvalues.

Le principe selon lequel on peut construire un sous-espace vectoriel en ne retenant que les meilleurs vecteurs propres, tout en conservant beaucoup d'informations utiles non redondantes, fait de la PCA un algorithme efficace et couramment utilisé en réduction de dimensionnalité et en reconnaissance de formes.

Nous déroulerons cet algorithme dans le Chapitre V.

III.3.3.3. Analyse Discriminante Linéaire (LDA) :

L'algorithme LDA est né des travaux de Belhumeur *et al* de la Yale Université (USA), en 1997 [R13]. Il est aussi connu sous le nom de Fisherfaces. Contrairement à l'algorithme PCA, l'algorithme LDA effectue une véritable séparation des classes. Pour pouvoir l'utiliser, il faut donc au préalable organiser la base d'images d'apprentissage en plusieurs classes : une classe par personne et plusieurs images par classe.

La LDA analyse les vecteurs propres de la matrice de dispersion des données, selon un critère mathématique qui minimise les variations entre les images d'un même individu (variations intra-classes) tout en maximisant les variations entre les images d'individus différents (variations interclasses). Cependant, pour des problèmes sous échantillonnés en reconnaissance de visage, c'est-à-dire lorsque le nombre d'individus à traiter est inférieur à la résolution de l'image, il est difficile d'appliquer la LDA qui peut alors faire apparaître des matrices de dispersions singulières (non inversibles). Afin de

contourner ce problème, certains algorithmes basés sur la LDA ont récemment été mis au point, le plus connu est l'algorithme R-LDA (Regularized LDA).

III.3.3.4. Machine à Vecteurs de Support (SVM) :

SVM (Support Vector Machines) est une nouvelle technique d'apprentissage statistique, proposée par Vapnik en 1995. Elle permet d'aborder des problèmes très divers comme le classement, la régression, la fusion,...etc. Depuis son introduction dans le domaine de la Reconnaissance de Formes (RdF), plusieurs travaux ont pu montrer l'efficacité de cette technique principalement en traitement d'image.

L'idée essentielle consiste à projeter les données de l'espace d'entrée (appartenant à des classes différentes) non linéairement séparables, dans un espace de plus grande dimension appelé espace de caractéristiques, de façon à ce que les données deviennent linéairement séparables [R14].

Dans cet espace, la technique de construction de l'hyperplan (dans un espace de dimension finie n , les hyperplans sont les sous-espaces vectoriel de dimension $n-1$) optimal est utilisée pour calculer la fonction de classement séparant les classes de sorte que :

- Les vecteurs appartenant aux différentes classes se trouvent de différents côtés de l'hyperplan.
- La plus petite distance entre les vecteurs et l'hyperplan (la marge) soit maximale.

III.3.3.5. Réseaux de neurones :

Les réseaux de neurones artificiels ou RNA sont des assemblages fortement connectés d'unités de calcul. Une image brute (ou prétraitée) de dimensions fixes constitue habituellement la source d'entrée des réseaux. Les dimensions doivent être établies au préalable car le nombre de neurones sur la couche d'entrée en dépend. Ainsi, plus les dimensions des images sont élevées, plus la complexité et le temps d'apprentissage augmentent. En effet, pour une image de dimensions 130×150 pixels, 19500 neurones seront requis sur la couche d'entrée, ce qui est énorme. L'apprentissage efficace d'un tel réseau est également douteux.

Le nombre de sorties du réseau dépend par ailleurs directement de la quantité d'individus à discriminer. L'apprentissage incrémental sera alors nécessaire et requerra des ajustements directs à l'architecture du réseau [R15].

Certains auteurs ont par ailleurs utilisé des variantes de la technique de base en modifiant les données d'entrée. Les coefficients de projections d'images dans un espace des visages (EigenFaces) peuvent par exemple être utilisés comme source d'information. Cette méthode peut évidemment être étendue aux coefficients de la DCT, de Fourier, ..., etc. L'avantage de ce modèle est sa robustesse face aux variations d'inclinaison et aux changements d'échelle. Cependant, l'utilisation d'exemples pour apprendre apporte le risque de ne pouvoir résoudre que des situations déjà rencontrées, c.à.d. un phénomène de sur-apprentissage qui spécialiserait le réseau uniquement sur les exemples connus sans généraliser.

III.3.3.6. Modèle surfacique du visage (3D) :

Cette méthode de reconnaissance repose sur l'utilisation d'un modèle tridimensionnel du visage. Pour que cette technique soit réellement efficace, une vue rapprochée du visage est nécessaire pour chacune des caméras impliquées dans l'acquisition. Une fois l'appariement des points réalisé, le modèle peut être normalisé et stocké dans la base de données. Lorsqu'un individu se présente devant les caméras, la même procédure s'applique, mais suivie d'une étape de comparaison.

En effet, le modèle à reconnaître doit être comparé à tous les modèles de la base de données, ce qui représente un travail énorme.

III.3.4. Test :

Pour évaluer les performances d'un système d'identification, on calcule le taux de reconnaissance du système.

On effectue N tests et on voit si la réponse du système (la personne la plus proche résultante de la comparaison entre le test et la base d'apprentissage) est juste, alors le taux est incrémenté et, à la fin, on aura le pourcentage $\text{taux}/100$ qui indique sur 100 tests combien de bonnes réponses sont retournées par le système. Ainsi plus ce taux s'approche de 100 plus le système est performant, sans oublier que nous sommes dans le cas d'un système temps réel qui doit être rapide, ce qui ajoute le facteur temps dans l'évaluation des systèmes biométriques d'identification.

Conclusion :

Nous avons dressé l'état des techniques de reconnaissance de visage. Nous allons voir dans le chapitre suivant le principe de la DCT.

Chapitre IV :
La transformée en cosinus
discrète

Introduction :

La pertinence de l'information n'étant pas aisée à apprécier dans le domaine direct de l'espace et du temps, nous chercherons donc à effectuer des changements de représentation qui permettraient de faciliter la séparation de l'information la plus pertinente.

Cette approche est celle des transformations orthogonales qui remplacent le signal d'origine par sa représentation suivant une base de fonctions, généralement par traitements de blocs.

Le but de la DCT est de diminuer la corrélation qui existe entre les valeurs des points d'une image, tout en minimisant la dimension par la sélection des coefficients les plus représentatifs, il en résulte un gain considérable en espace mémoire et en temps de calcul.

IV.1. Formule mathématique de la DCT :

La DCT appartient à une classe d'opérations mathématiques semblable à la Transformée de Fourier (*DFT*), elle permet un changement de domaine d'étude, tout en gardant exactement la même fonction étudiée. Dans notre cas, nous étudions une image, c'est-à-dire une fonction à 2 dimensions : X et Y, indiquant les coordonnées spatiales du pixel. Dans le cas d'une image couleur, nous devons considérer indépendamment 3 fonctions, pour chacun des canaux RGB¹.

La DCT d'un signal discret à deux dimensions est définie comme suit :

Soit $f(x,y)$ la fonction décrivant une séquence de longueur $N*N$. La transformée en cosinus discrète 2-D de la séquence $f(x,y)$ notée $C(u,v)$ est la suivante :

$$C(u, v) = a(u)a(v) \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} f(x, y) \cos\left(\frac{\pi(2x+1)u}{2N_1}\right) \cos\left(\frac{\pi(2y+1)v}{2N_2}\right) \quad (1)$$

$$u = 0, \dots, N_1 - 1 ; v = 0, \dots, N_2 - 1$$

De même, nous définissons la transformée en cosinus discrète inverse 2-D comme suit :

¹ RGB: Red (rouge), Green (vert), Blue (bleu).

$$f(x, y) = \sum_{u=0}^{N_1-1} \sum_{v=0}^{N_2-1} a(u)a(v) C(u, v) \cos\left(\frac{\pi(2x+1)u}{2N_1}\right) \cos\left(\frac{\pi(2y+1)v}{2N_2}\right) \quad (2)$$

$$x = 0, \dots, N_1 - 1 ; y = 0, \dots, N_2 - 1$$

$$\text{Avec } a(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases} \quad \text{pour (1) et (2)}$$

La composante $C(0,0)$ est appelée composante *DC* et elle représente la moyenne de l'image ; quant aux autres composantes ($C(u,v) / u \neq 0$ ou $v \neq 0$), elles sont appelées les composantes *AC*.

En ignorant les composantes $a(u)$, $a(v)$ et $f(x,y)$ de l'équation (1), les fonctions de bases sont définies par la partie restante : $C(u) = \cos\left(\frac{\pi(2x+1)u}{2N_1}\right) \cos\left(\frac{\pi(2y+1)v}{2N_2}\right)$

Nous remarquons que les fonctions de bases 2-D augmentent en fréquence au fur et à mesure que nous augmentons les paramètres u et v .

IV.2. Les caractéristiques de la DCT :

Cette section décrit les propriétés de la DCT appliquées aux traitements d'image :

- Compactage de l'énergie :

La DCT montre un excellent pouvoir de compactage d'énergie pour des images fortement corrélées. Elle assure la concentration de l'énergie dans les basses fréquences du domaine transformé, donc nous pouvons représenter l'intégralité de l'information de l'image sur très peu de coefficients.



Figure IV.1 : La concentration d'énergie par la DCT

- Décorrélation :

Le principal avantage de la transformation d'images est d'enlever la redondance entre les pixels voisins. Cette redondance est due au fait que les images naturelles sont fortement corrélées (les valeurs des pixels voisins sont presque identiques).

Cette décorrélation est expliquée par l'orthogonalité de ses fonctions de bases et, par conséquent, les composantes de la DCT sont indépendantes, ce qui nous permet de les traiter individuellement.

- Séparabilité :

La fonction de la DCT 2-D peut s'écrire sous la forme :

$$C(u, v) = a(u)a(v) \sum_{x=0}^{N_1-1} \cos\left(\frac{\pi(2x+1)u}{2N_1}\right) \sum_{y=0}^{N_2-1} f(x, y) \cos\left(\frac{\pi(2y+1)v}{2N_2}\right) \quad (3)$$

La DCT permet de séparer le calcul de $C(u, v)$, en utilisant la propriété de séparabilité.

Cette propriété a l'avantage que $C(u, v)$ peut être calculé dans deux étapes par les opérations DCT 1-D successives sur des rangées et des colonnes d'une image. Cette idée est graphiquement illustrée sur le schéma suivant :

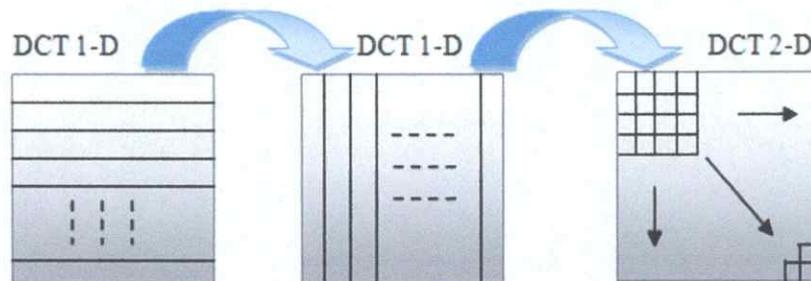


Figure IV.2 : Séparabilité de la DCT 2-D

- Symétrie :

Un autre regard sur les opérations de ligne et de colonne dans l'équation (3) indique que ces opérations sont fonctionnellement identiques. Une telle transformation s'appelle une transformation symétrique, qui peut être exprimée sous la forme :

$$T = A f A$$

Où A est une matrice de transformation symétrique ($N*N$) avec $a(i, j)$ donnée par :

$$a(i, j) = \alpha(j) \sum_{j=0}^{N-1} \cos\left(\frac{\pi(2j+1)i}{2N}\right)$$

C'est une propriété extrêmement utile puisqu'elle implique que la matrice de transformation peut être précalculée, il en résulte une amélioration de l'efficacité de calcul.

- Indépendance des données :

En observant de près l'équation de la DCT, nous pouvons déduire que *les fonctions de base ne résultent pas de l'image $f(x,y)$* ; nous pouvons alors affirmer que cette transformée est indépendante des données ; ainsi le précalcul des fonctions de bases est possible, ce qui donne un gain considérable en temps de calcul.

IV.3. DCT vs KLT :

Il existe une transformation unitaire unique capable de décorrélérer complètement l'ensemble des pixels d'une image, c'est la transformée de Karhunen-Loève (KLT).

Cette transformée possède une seconde propriété intéressante, elle assure la meilleure concentration de l'énergie dans le domaine transformé, mais la DCT l'approche asymptotiquement comme le montre ce graphe :

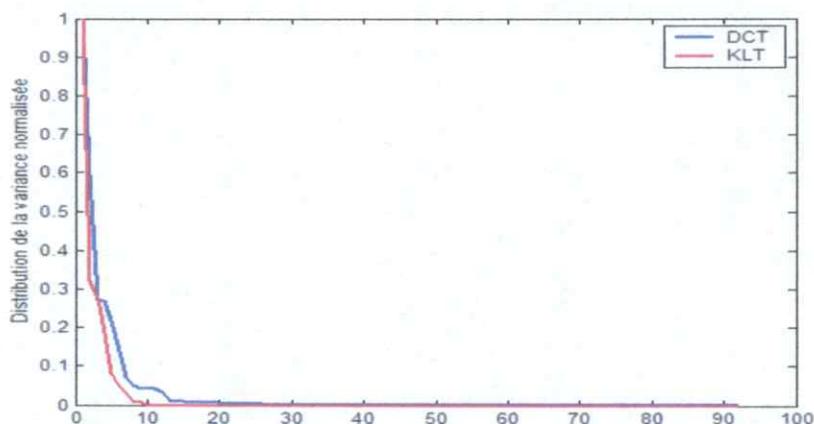


Figure IV.3 : DCT vs KLT en variance

Cependant, la KLT présente un inconvénient, son temps de calcul qui ne permet pas son utilisation en pratique ; la KLT est dépendante des données, par contre la DCT

dont les propriétés sont illustrées ci-dessus montrent qu'elle est symétrique, donc ses fonctions de base sont précalculées, ce qui réduit considérablement son temps de calcul.

Nous optons donc pour la DCT parce que c'est une bonne approximation de la KLT en matière de compression, en plus elle est plus rapide à l'exécution.

IV.4. DCT vs DFT :

Comme la DCT, la DFT (transformée de Fourier) est une transformation séparable et symétrique, et de ce fait, elle permet des réalisations rapides.

Par ailleurs, elle permet de bonnes concentration et décorrélation d'énergie. Cependant, la DFT est une transformation complexe (le noyau est une exponentielle qui génère des coefficients complexes) et nécessite donc un codage et un traitement pour les paramètres phase et amplitude. En outre la périodicité implicite de la DFT provoque les discontinuités de frontière qui ont comme conséquence l'apparition des coupures et donc elle ne permet pas un codage précis. D'où le choix de la DCT.

IV.5. Types de DCT :

IV.5.1. La DCT par bloc 8x8 :

IV.5.1.1. Définition :

Pour des raisons de performance et de complexité de calcul, la DCT est souvent utilisée sur des blocs de taille 8x8. Voici un schéma illustratif expliquant les étapes de l'application de la DCT par bloc sur une image :

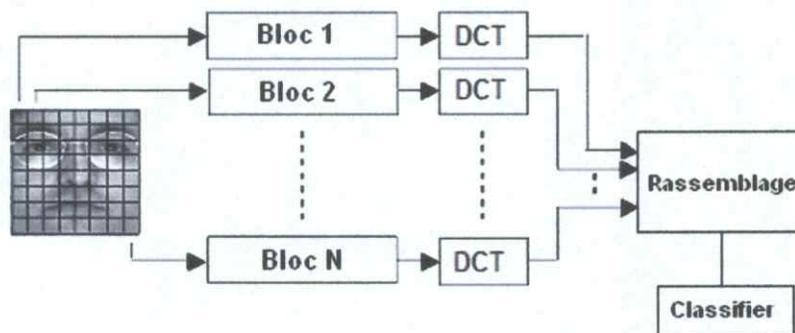


Figure IV.4 : Application de la DCT par bloc

Cette configuration est utilisée dans les standards JPEG² et MPEG³, utilisant une DCT 2-D sur des blocs de pixels de taille 8x8, en plus il a été prouvé que l'application de la DCT sur les blocs 8x8 est un très bon compromis entre la complexité de calcul et le taux de compression.

IV.5.1.2. La sélection des coefficients :

L'information est essentiellement portée par les coefficients basses fréquences et donc elle peut être préservée par un nombre très petit de coefficients. Par conséquent, la plupart des coefficients de la DCT peuvent être ignorés, ce qui réduit la dimensionnalité.

Il existe plusieurs méthodes de sélection des coefficients DCT, la plus utilisée est la méthode appelée *ZIGZAG*, elle consiste en un parcours zigzag des coefficients de l'image DCT.

Nous présentons la matrice qui donne l'ordre des coefficients DCT d'un bloc 8x8 lors de la sélection en Zigzag :

$$\text{Ordre} = \begin{pmatrix} 0 & 1 & 5 & 6 & 14 & 15 & 27 & 28 \\ 2 & 4 & 7 & 13 & 16 & 26 & 29 & 42 \\ 3 & 8 & 12 & 17 & 25 & 30 & 41 & 43 \\ 9 & 11 & 18 & 24 & 31 & 40 & 44 & 53 \\ 10 & 19 & 23 & 32 & 39 & 45 & 52 & 54 \\ 20 & 22 & 33 & 38 & 46 & 51 & 55 & 60 \\ 21 & 34 & 37 & 47 & 50 & 56 & 59 & 61 \\ 35 & 36 & 48 & 49 & 57 & 58 & 62 & 63 \end{pmatrix}$$

En suivant cet ordre, on construit le vecteur de coefficients DCT ordonné de manière descendante suivant l'énergie portée par chacun d'eux.

IV.5.1.3. Chevauchement des blocs 8x8 :

Pour éviter les effets de discontinuité entre les blocs DCT, on peut faire chevaucher les blocs dans l'image horizontalement et verticalement. Le degré de chevauchement est exprimé en pourcentage, si on a un chevauchement de 50% alors 50% des pixels formant un bloc i vont construire 50% du bloc $i+1$, ainsi les blocs voisins horizontalement et verticalement se ressembleront à 50%.

² Joint Picture Expert Group : norme informatique internationale de compression des images fixes.

³ Motion Picture Expert Group : norme informatique internationale de compression des images animées avec son.

IV.6. La DCT dans la reconnaissance de visage :

Algorithme : on applique la DCT pour chaque image et on sélectionne les composantes selon les types de DCT (tri en zigzag ou matrice).

Dans ce qui suit nous présentons un schéma qui montre comment fonctionne un système de reconnaissance par la DCT :

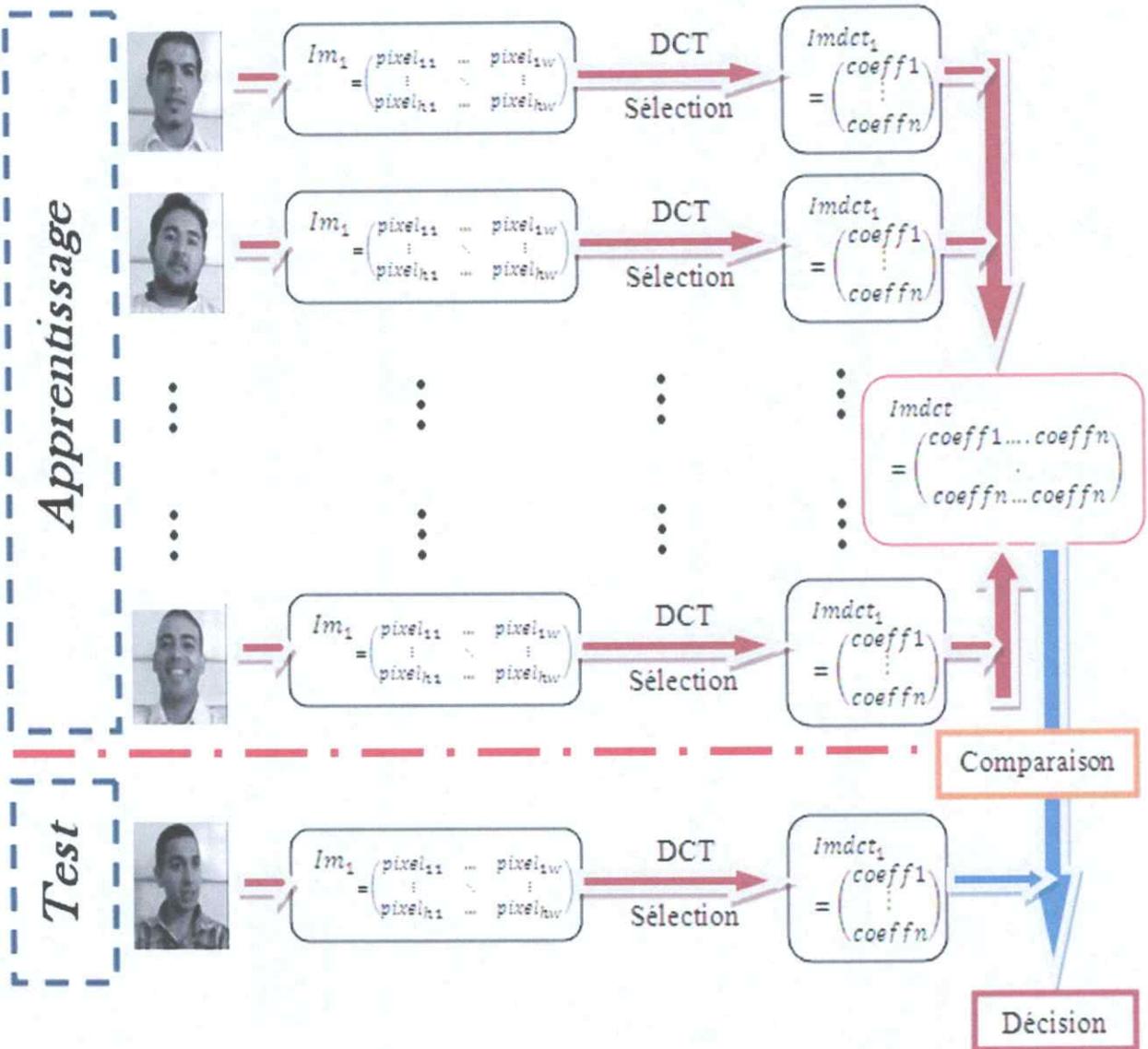


Figure IV.6 : reconnaissance par DCT.

Conclusion :

À travers ce chapitre, nous avons vu que la DCT est une transformation très rapide à l'exécution et elle exploite au mieux les redondances inter pixels pour rendre une excellente décorrélation de la plupart des images naturelles.

De plus, la DCT concentre l'énergie dans les régions de basses fréquences, par conséquent, une partie du contenu à hautes fréquences peut être ignorée sans dégradation significative de l'information. Ainsi, après avoir vu en détail le principe de la DCT et souligné ses avantages, nous allons présenter la PCA et voir son fonctionnement et ce qu'elle a de particulier par rapport aux autres transformées.

Chapitre V :
L'Analyse en Composante
Principale

Généralités :

Voici une première manière de voir l'Analyse en Composantes Principales (ACP ou PCA¹ pour Principal Component Analysis) : on dispose d'un nuage de points dans un espace de dimension élevée, duquel on ne peut extraire l'information utile à l'analyse, l'ACP nous donnera un sous-espace de dimension raisonnable, de sorte que la projection sur ce dernier retienne le plus d'informations possibles et que le nuage résultant soit le plus dispersé possible.

Le contexte dans lequel on applique cette technique est le suivant : on observe sur n individus p variables explicatives X_1, X_2, \dots, X_p quantitatives, présentant des relations multiples qu'on veut analyser. Pour pouvoir interpréter ces données, il est préférable de les dissocier et de réduire le nombre de caractères.

La PCA permet de résumer l'information apportée par ces p variables en M composantes indépendantes ($M \leq P$) en la détruisant le moins possible.

La PCA est aussi connue sous le nom de Transformée de Karhunen-Loève (*KLT*), elle se base sur la matrice de covariance (corrélation dans le cas de variables centrées réduites) des variables pour trouver le sous-espace optimal pour représenter les données. Ainsi les composantes principales obtenues sont la projection des variables (combinaison linéaire) dans le nouvel espace réduit.

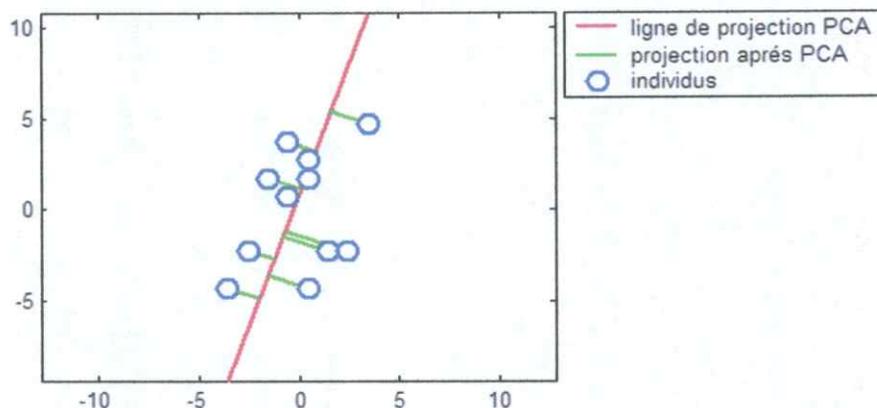


Figure V.1 : Exemple de projection suivant PCA

¹ Dans ce qui va suivre, c'est l'acronyme PCA qui revient le plus souvent.

Il existe plusieurs variantes de la PCA, parmi lesquelles nous pouvons citer :

- (2D) PCA (two-dimensional principal component analysis)
- (2D)² PCA (directionnel principal component analysis)
- KPCA (kernel principal component analysis)
- Grain PCA
- ICA (independent component analysis)

Dans le cadre de notre recherche, nous n'allons définir que la PCA et la (2D)PCA.

V.1. Principe de la PCA:

On considère un vecteur X représentant une distribution de n données de dimension p (p variables explicatives) :

$$X = [X_1 \ X_2 \ \dots \ X_i \ \dots \ X_n].$$

Nous désirons représenter le plus fidèlement possible cette distribution dans un espace E de dimension k ($k \leq p$) de telle sorte à avoir des données indépendantes entre elles (matrice de covariance diagonale).

Soit $Y = [Y_1 \ Y_2 \ \dots \ Y_n]$ la projection de X dans le nouveau sous-espace, alors $Y = W^T X$ (W^T est la matrice transposée de W)

Ainsi le problème est de trouver W tel que la matrice de covariance de Y soit diagonale (assurer l'indépendance des données) et avoir une concentration des données pour permettre la réduction de dimensionnalité.

Notons par Σ_x la matrice de covariance de X ($\Sigma_x = X X^T$ avec X centré à sa moyenne), Σ_y celle de Y et par Λ la matrice des valeurs propres de Σ_x .

La PCA résout ce problème par la diagonalisation de Σ_x , ce qui revient à calculer les vecteurs propres et valeurs propres de cette dernière et donc $\Sigma_x W = \Lambda W$.

La matrice de projection W sera celle des vecteurs propres de Σ_x .

$$\begin{aligned} \text{On a : } \dim(X) &= (p \cdot n). & (\dim(X) \text{ désigne la dimension de } X) \\ \dim(Y) &= (k \cdot n). & \text{ Avec } k = \text{rang}(\Sigma_x = X X^T) = \min(n, p) \\ \dim(W) &= (p \cdot k). \\ \dim(\Lambda) &= (k \cdot k). \end{aligned}$$

Et donc : $\Sigma_y = Y.Y^T = W^T X X^T W = W^T \Sigma_x W$. (1)

Σ_y est la rotation de Σ_x par W .

Nous avons aussi : $\Sigma_x W = \Lambda W \Rightarrow \Sigma_y = W^T \Lambda W = \Lambda W W^T = \Lambda$, en effet Σ_y est la matrice des valeurs propres de Σ_x et donc **diagonale**, ainsi la variance de chaque composante principale est maximisée à la valeur propre correspondante et elle est indépendante des autres $k-1$ composantes restantes.

V.2. Quelques propriétés des PCA :

V.2.1. Concentration de l'information et réduction de dimensionnalité :

Soit le vecteur $x = (x_1, \dots, x_p)$ de matrice de covariance Σ_x avec les paires de valeurs propres, vecteurs propres $(\lambda_1, e_1), \dots, (\lambda_p, e_p)$ et $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$, alors on a :

La variance totale de la population est égale à :

$$\sum_{i=1}^p \text{Var}(X_i) = \text{trace}(\Sigma_x) = \text{trace}(W \Lambda W^T) = \text{trace}(\Lambda) = \lambda_1 + \lambda_2 + \dots + \lambda_p. \quad (2)$$

Ainsi la proportion de la variance totale expliquée par la k -ième composante est égale à :

$$\frac{\lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_p} \quad (3)$$

Nous pouvons de cette façon réduire la dimensionnalité en ne gardant que les vecteurs propres correspondant aux valeurs propres suffisamment grandes. Il existe plusieurs manières de désigner le nombre de valeurs propres nécessaires pour une réduction sans perte d'information considérable, nous pouvons citer par exemple celle qui prend les m valeurs propres tel que :

$$\frac{\lambda_1 + \lambda_2 + \dots + \lambda_m}{\lambda_1 + \lambda_2 + \dots + \lambda_p} \geq 95\% \quad (4)$$

Nous pouvons voir la concentration d'énergie sur m caractères, car la décroissance des valeurs λ_i est généralement exponentielle ($m \ll p$).

V.2.2. Indépendance de l'information :

Σ_y étant diagonale, $cov(y_i, y_j) = 0$ si $i \neq j$, cette condition définit l'indépendance. et de ce fait, les données étant indépendantes, leur classification sera plus aisée.

V.2.3. PCA et direction de la variance :

Les λ_i représentent les variances des composantes principales et leurs valeurs déterminent la direction de la projection grâce aux vecteurs propres correspondants, alors la population en entrée est projetée en direction de la plus grande variance pour obtenir la nouvelle population.

V.3. PCA classique dans la reconnaissance de visage :

La PCA a été introduite dans la reconnaissance de visage par MA. Turk et MP. Pentland, en 1991. Son but principal est d'extraire l'information pertinente de l'image visage et la coder le plus efficacement possible puis la comparer avec la base des modèles codés de la même manière. Une approche simple pour extraire l'information utile de l'image est de capturer les variations dans un ensemble de visages et utiliser cette information pour coder et comparer les visages entre eux.

En termes mathématiques, il s'agit de calculer les composantes principales (Eigenfaces) d'une distribution de visages à partir des vecteurs propres de la matrice de covariance de l'ensemble des visages et de les ordonner suivant leurs valeurs propres correspondantes. Chaque vecteur propre suivant sa valeur propre capturera une grande ou petite variation des visages de la distribution étudiée et l'ensemble de ces vecteurs caractérise les variations totales des images.

Ainsi, ces vecteurs propres représenteront la base de projection des visages dans l'espace propre, soit dans l'apprentissage pour produire les modèles ou dans le test pour pouvoir classifier en comparant le modèle test avec ceux de la base d'entraînement.

Algorithme :

On considère un ensemble de M images visages de taille (m, n) , chacune d'elles est représentée sous forme vectorielle de taille $(m*n, 1)$. Par analogie à ce que nous avons défini précédemment, X de taille $(m*n, M)$, représente la matrice des images d'apprentissage du système tel que chaque colonne est une image et chaque ligne est

une valeur de pixel (qui correspond aux variables explicatives dans la PCA) dans l'image correspondante.

Nous décomposons l'algorithme en plusieurs étapes que nous présentons ici :

- Apprentissage :

- Centrer les valeurs de X en faisant $X=X-\mu$ tel que μ est le vecteur moyen de X , car la covariance est définie pour des données centrées.

CAS $m*n > M$:

- Calculer $\sum_x \leftarrow X^T X$ pour gagner en calcul car la taille de \sum_x est $n*n$ au lieu de $m*m$, nous trouvons les mêmes composantes que dans le cas normal en faisant intervenir l'algèbre linéaire.
- Calculer la matrice des vecteurs propres W ordonnés suivant la matrice des valeurs propres Λ , elle même triée par ordre décroissant.
- Pour revenir au cas normal, nous effectuons les opérations suivantes : $W \leftarrow (W.X)$ puis nous normalisons chaque vecteur propre (colonne de W) en le divisant par sa norme euclidienne et nous obtenons ainsi la matrice de projection W .

CAS $m*n \leq M$:

- Calculer $\sum_x \leftarrow X X^T$.
- Calculer la matrice des vecteurs propres W ordonnés suivant la matrice des valeurs propres Λ , elle même triée par ordre décroissant.
- Projeter les images dans le nouvel espace qu'on appellera espace propre en faisant : $Y \leftarrow W^T X$, Y représente les images (les modèles) dans l'espace propre.

- Test :

- Une image test se présente dans le système, on la transforme en un vecteur de taille $(m*n, 1)$ qu'on note $Test$.
- Projeter l'image dans l'espace propre : $Test-propre \leftarrow W^T Test$.
- Comparer le $Test-propre$ avec les modèles obtenus dans l'apprentissage. La comparaison se fait par le calcul des distances entre les vecteurs de modèles et $Test-propre$ (classification).

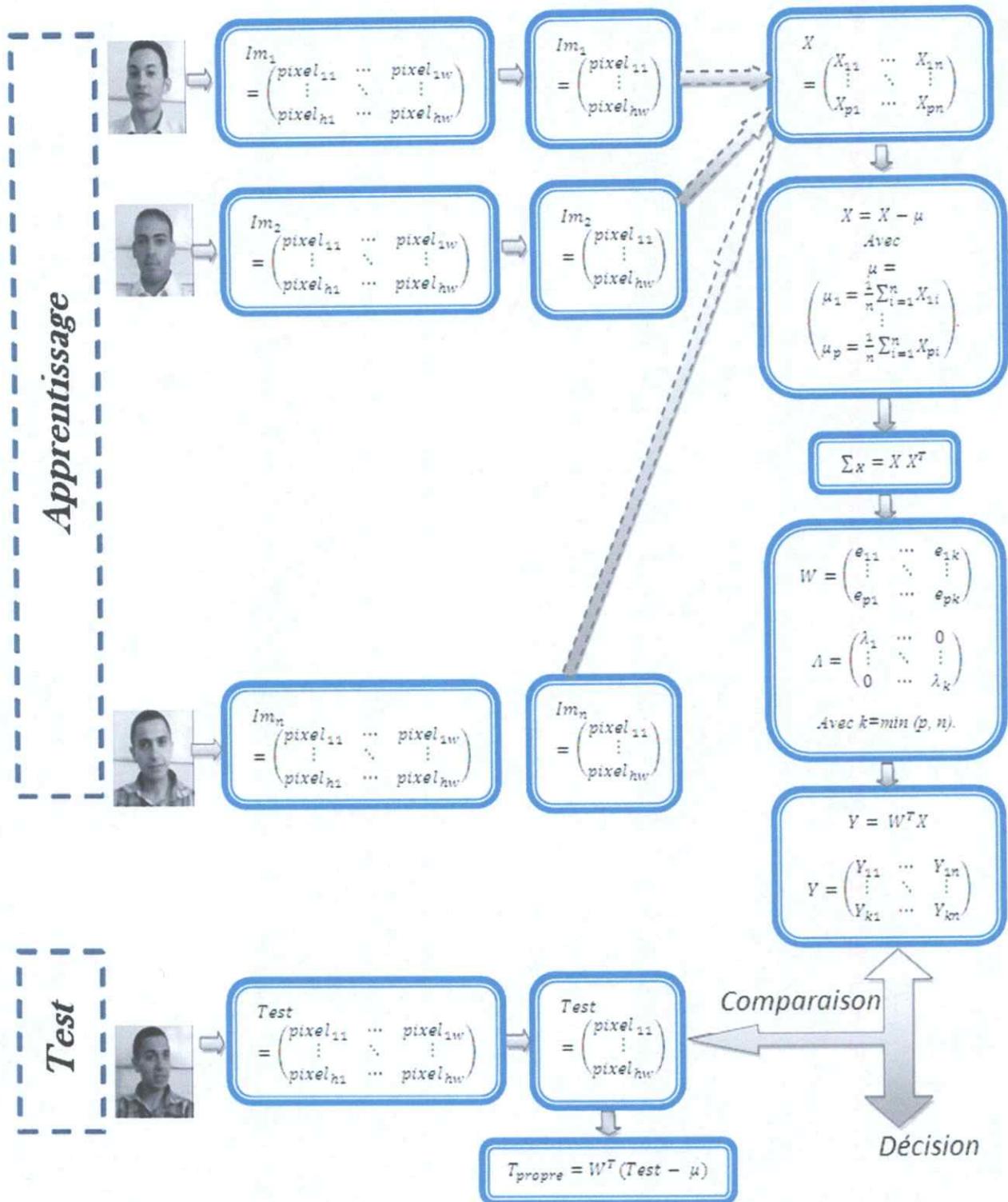


Figure V.2 : Processus de reconnaissance par PCA

V.4. La (2D) PCA dans la reconnaissance de visage :

L'analyse en composante principale bidimensionnelle (2D) PCA a été proposée par J. Yang et al. en 2004 [R16] pour la représentation et l'identification de visage. L'idée principale de la (2D)PCA est qu'elle est basée sur les matrices 2D en opposition à la PCA standard qui est basé sur les vecteurs 1D. Dans cette approche le calcul de la matrice de covariance se fait à partir des matrices 2D des images d'apprentissage ainsi sa taille est plus petite que celle de la PCA. En conséquence, il est plus facile et plus rapide d'évaluer la matrice de covariance et le temps de calcul des vecteurs propres correspondants est diminué.

Algorithme :

Nous considérons un ensemble de M images visages de taille (m,n) notées A_i ($i=1, \dots, M$), nous décomposons l'algorithme en plusieurs étapes que nous présentons ici :

- Apprentissage :

- Calculer la moyenne des images apprentissage \bar{A} de la manière suivante :

$$\bar{A} \leftarrow \frac{1}{M} \sum_{i=1}^M A_i$$

- Calculer la covariance :

$$\Sigma_A \leftarrow \frac{1}{M} \sum_{i=1}^M (A_i - \bar{A})^T (A_i - \bar{A})$$

Σ_A est de taille $n \times n$

- Calculer la matrice des vecteurs propres W ordonnés suivant la matrice des valeurs propres Λ , elle même triée par ordre décroissant.
- Projeter les images dans le nouvel espace propre, $Y_i \leftarrow A_i W$ (w est de taille (n,d) tel que $d \leq n$) correspondant au modèle d'apprentissage de la $i^{\text{ème}}$ image A_i . $\text{modèles}_{\text{apprentissage}} = [Y_1, \dots, Y_M]$ est le modèle d'apprentissage dans l'espace propre de la (2D)PCA.

- Test:
 - Une image test se présente dans le système, on la projette dans l'espace propre : $modèle_{test} \leftarrow image_{test}W$
 - Calculer la distance entre le $modèle_{test}$ et les $modèles_{apprentissage}$

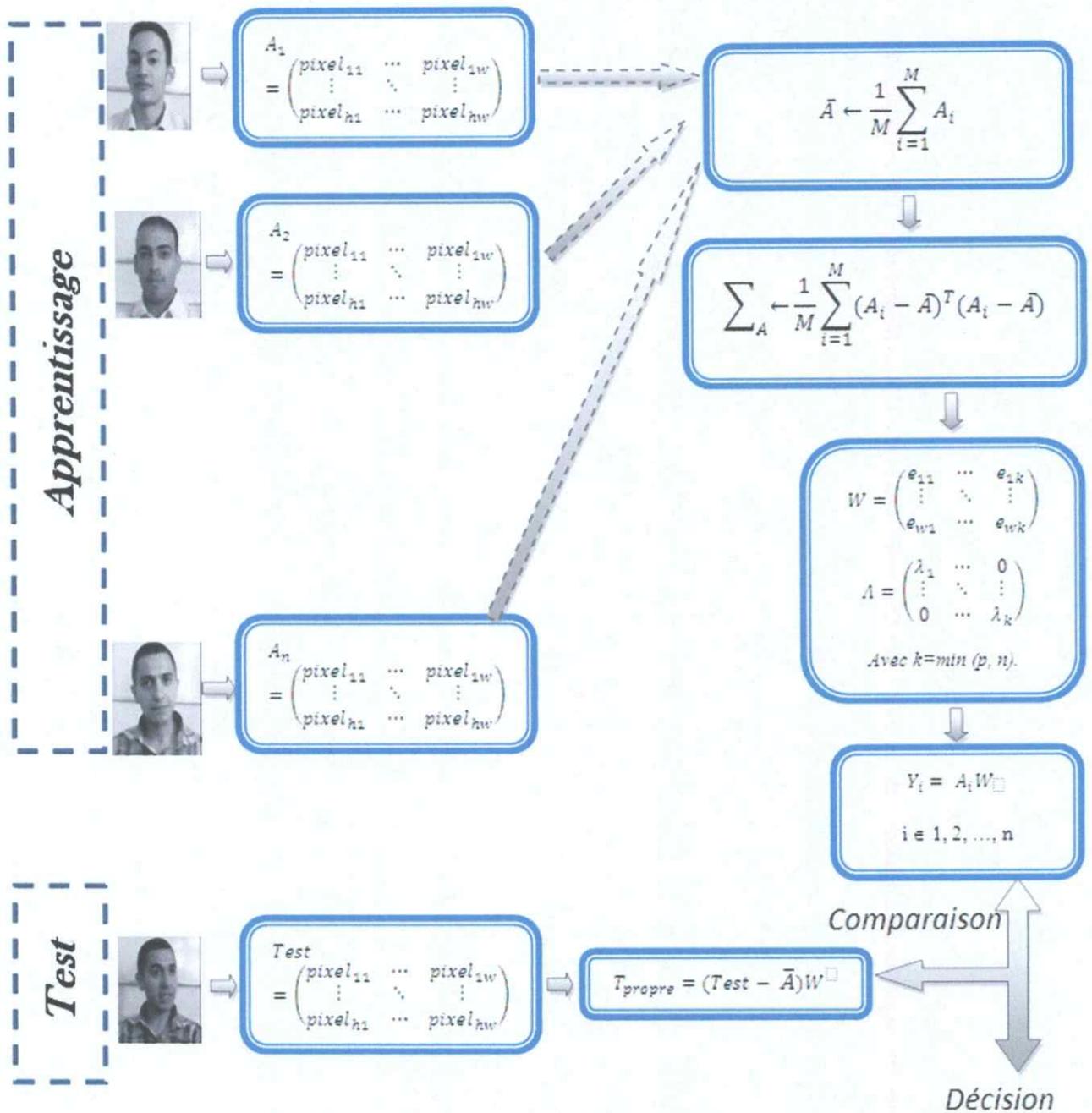


Figure V.3 : Processus de reconnaissance par (2D)PCA

IV.5. PCA dans le domaine DCT :

La complexité du calcul engendrée par la PCA, surtout quand la résolution des images et de la taille de l'échantillon d'apprentissage, est grande. Ce qui a poussé les chercheurs à intégrer une étape qui pourrait diminuer le temps de calcul pour parvenir à un vrai système de reconnaissance de visage, car le temps de réponse est très important dans ce cas. L'une des solutions préconisées par le CDTA² est d'utiliser le vecteur de coefficient DCT (ou la matrice des coefficients DCT Globale pour la (2D)PCA) de l'image au lieu d'utiliser image elle-même (cf. Figure V.4).

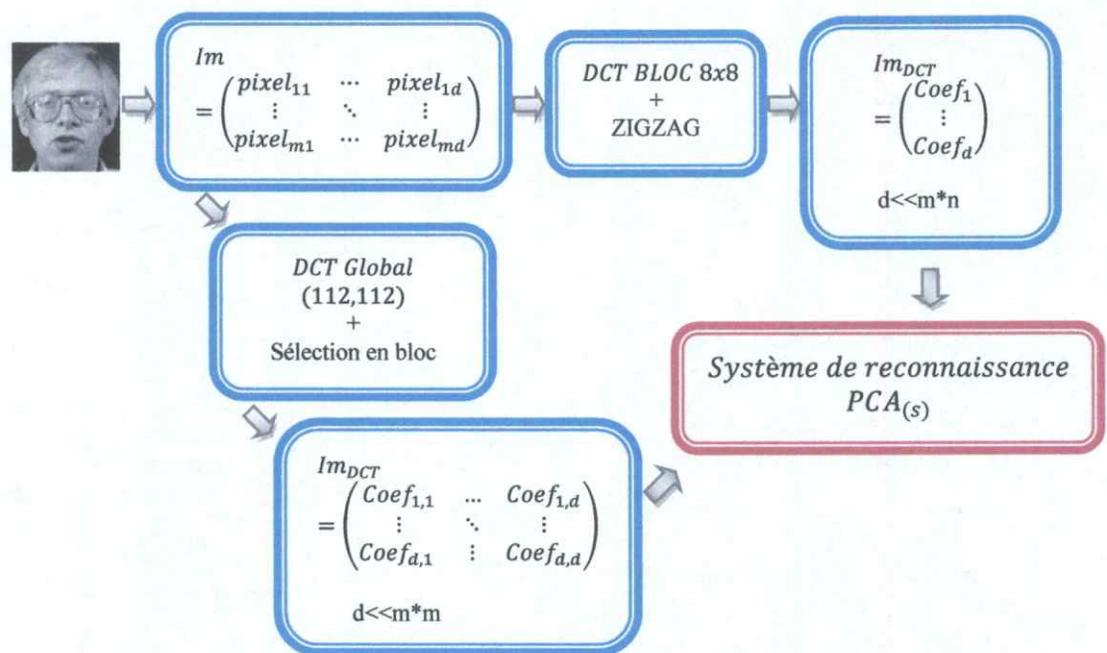


Figure V.4: la PCA dans le domaine DCT

² CDTA : Centre de Développement des Technologies Avancées

Conclusion :

Comme nous venons de le voir, la PCA est une méthode statistique pratique dans le domaine de la reconnaissance de visage. Elle projette les données en direction de la plus grande variance, pour donner des composantes indépendantes et ce, afin d'obtenir une meilleure discrimination.

Mais les limites de la PCA, dans le domaine de la reconnaissance de visage, sont ressenties quand la direction de la plus grande variance ne sépare pas les classes (personnes) et, en cas d'ajout d'une nouvelle personne à la base de données, le système doit être recyclé (refaire l'apprentissage) car les résultats de la PCA (la matrice W) dépendent de l'ensemble total des images de l'échantillon d'apprentissage. Nous allons présenter dans le chapitre qui suit la DiaPCA.

Chapitre VI :
L'Analyse Diagonale en
Composantes Principales

Introduction :

L'analyse en composantes principales (PCA) est l'une des méthodes les plus utilisées dans la reconnaissance de visage. Dans l'identification de visage basée sur la PCA, les images 2D sont transformées en vecteurs colonnes 1D. Le calcul de la matrice de covariance à base de ces vecteurs est difficile à cause de la grande taille des vecteurs 1D et le nombre important d'échantillons d'apprentissage. En générale, le calcul des vecteurs propres d'une grande matrice de covariance prend beaucoup de temps.

Récemment, une nouvelle technique appelée l'analyse en composantes principales bidimensionnelle (2DPCA) a été proposée. Son idée principale est de calculer la matrice de covariance à partir des matrices 2D des images d'apprentissage ainsi sa taille est plus petite que la matrice de covariance de la PCA. En conséquence, il est plus facile et plus rapide d'évaluer la matrice de covariance et le temps de calcul des vecteurs propres correspondants est diminué.

L'analyse en composantes principales diagonale a été proposée D. Zhang et al. en 2006 afin d'améliorer la 2DPCA puisque ses vecteurs propres reflètent seulement les variations entre les lignes de l'image, tandis que les variations entre les colonnes sont omises. Dans la DiaPCA, l'image diagonale de chaque visage est calculée en employant un des deux algorithmes proposés dans [R17], ensuite une matrice de covariance diagonale est calculée, puis les vecteurs propres sont déterminés. Ainsi, la corrélation entre les variations des lignes et ceux des colonnes est gardée.

VI.1. Algorithme de diagonalisation des images :

Dans l'article [R17], les auteurs ont défini deux manières de diagonaliser une image de taille $m \times n$, *cette diagonalisation est une transformation qui est effectuée sur les images d'apprentissage afin d'obtenir une autre représentation des images*. Elle est effectuée comme suit :

1. Si m est plus petit ou égal à n , employer la méthode illustrée ci-dessous :

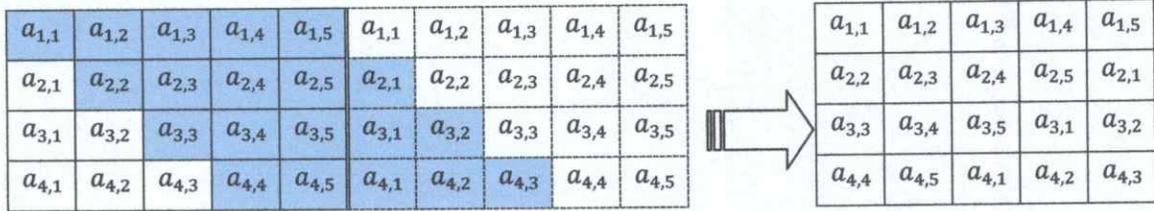


Figure VI.1 : Construction d'une image diagonale dans le cas où les lignes sont supérieures aux colonnes.

2. Si m est plus grand ou égal à n , employer la méthode illustrée ci-dessous :

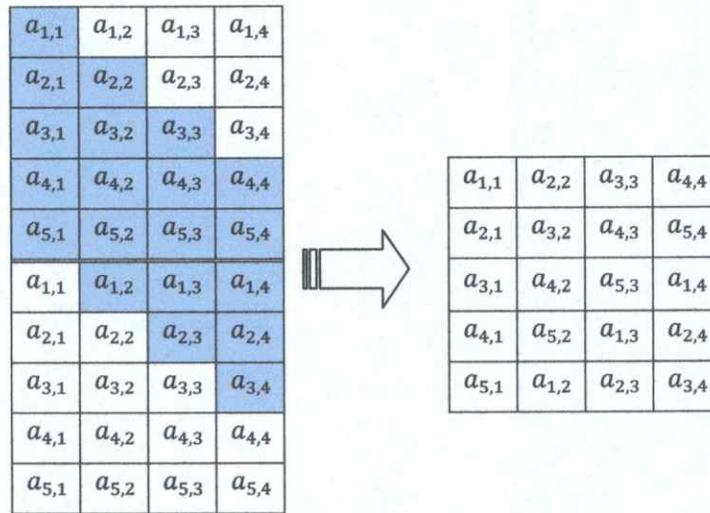


Figure VI.2 : Construction d'une image diagonale dans le cas où les colonnes sont supérieures aux lignes

Nous avons apporté quelques modifications à la diagonalisation proposée par les deux auteurs et ce, afin de réduire les calculs et donc le temps de calcul :

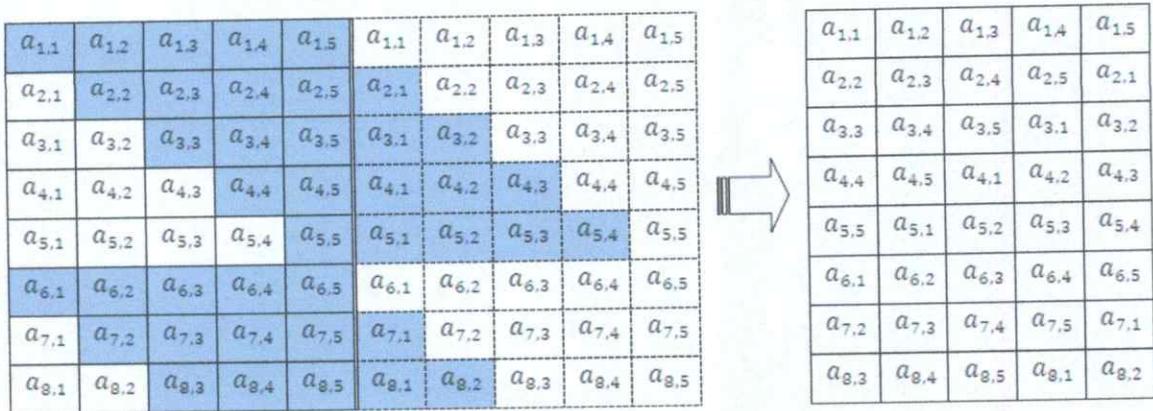


Figure VI.3 : Construction d'une image diagonale.

Exemple sur une image

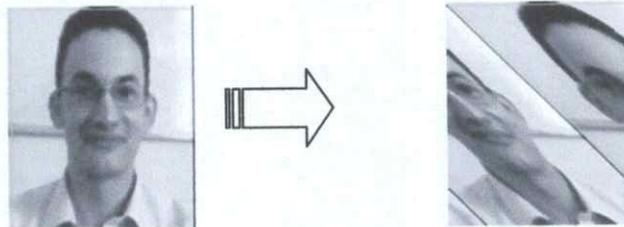


Figure VI.4 : Application de la diagonale sur une image.

VI.2. Diagonale (2D) PCA :

L'analyse en composantes principales diagonale (Diagonale (2D) PCA) a été proposée D. Zhang et al. en 2006, cette méthode vient d'une observation sur la (2D) PCA, [R17]. Cette dernière traite l'information, exclusivement, entre les rangées, ainsi les régions du visage comme les yeux, le nez, etc...ne peuvent pas être découvertes. En transformant les images d'origine du visage en images diagonales, la DiaPCA intègre simultanément l'information des lignes et des colonnes dans les images ; elle trouve, alors, de l'information utile de bloc ou de structure pour l'identification.

Supposons qu'il y a M images ($m \times n$) d'apprentissage notées $A_k (k = 1, \dots, M)$. Pour chaque image A_k , on définit son image diagonale correspondante B_k telle que définie plus haut (Fig. VI.3).

En se basant sur les visages diagonaux, nous définissons la matrice de covariance diagonale G :

$$G = 1/M \sum_{k=1}^M (B_k - \bar{B})^T (B_k - \bar{B}) \quad \text{telq } \bar{B} = 1/M \sum_{k=1}^M B_k \quad (1)$$

Les vecteurs projectifs X_1, \dots, X_d sont obtenus en calculant les valeurs et vecteurs propres correspondant aux d plus grandes valeurs propres de $G(n \times n)/d < n$.

Soit $X = [X_1, \dots, X_d]$, la matrice de projection, le calcul des nouveaux composants C (modèles) est le suivant :

$$C_k = A_k X \quad (2)$$

Ayant une image de test T , nous la projetons dans le nouvel espace propre diagonal selon la formule (2) $C = TX$, puis nous recherchons le voisin le plus proche pour l'employer comme classificateur. La distance entre C et C_k est définie comme suit :

$$\begin{cases} d(c, c_k) = |c - c_k| = \sum_{j=1}^d \left(\sum_{i=1}^K (c^{(i,j)} - c_k^{(i,j)})^2 \right)^{1/2} & (3) \\ \text{où} \\ d(c, c_k) = \|c - c_k\| = \sqrt{\sum_{i=1}^K \sum_{j=1}^d (c^{(i,j)} - c_k^{(i,j)})^2} & (4) \end{cases}$$

Tel que Eq(3) et (4) représentent la distance de Yang et de Frobenius respectivement. Ces 2 métriques sont utilisées pour le calcul de distance entre 2 matrices, puisque dans la Diagonal (2D)PCA nous travaillons avec des matrices contrairement à la PCA qui est basé sur les vecteur 1D. les détails de ces métriques seront présentés dans l'annexe.

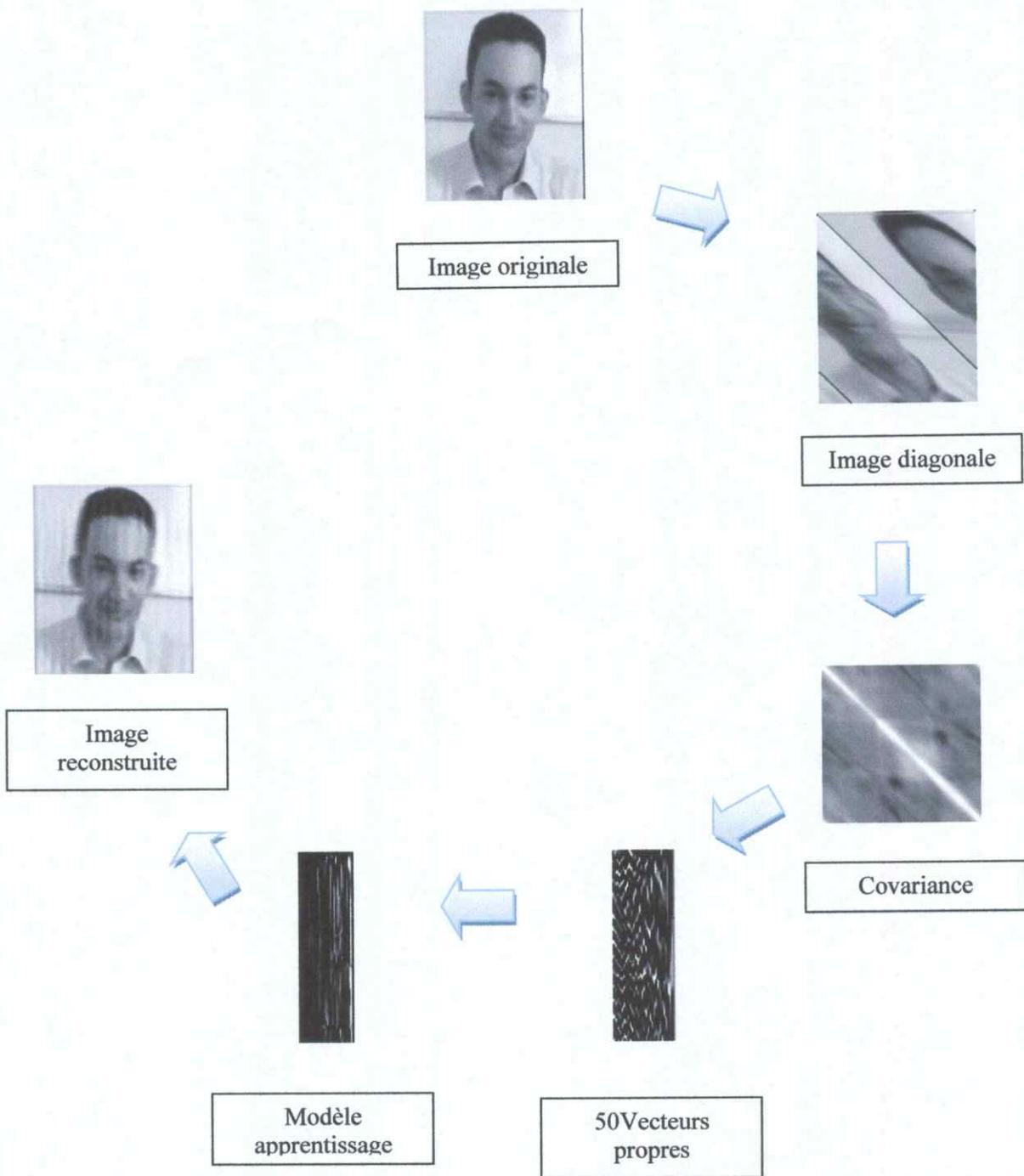


Figure VI.5 : Application de la reconnaissance par la diagonale PCA.

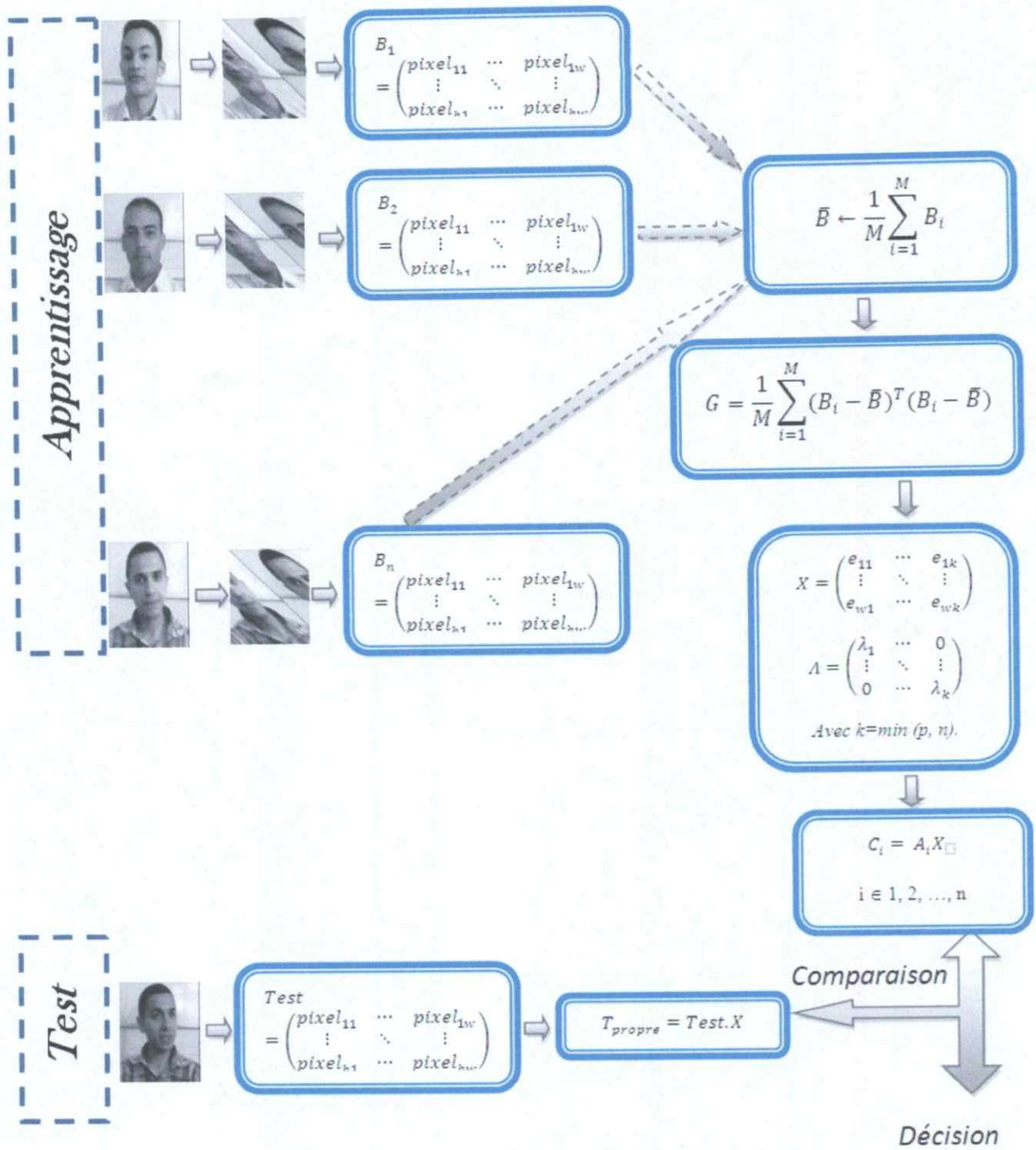


Figure VI.6 : Processus de reconnaissance par la Diagonale (2D) PCA

VI.3. Diagonale PCA :

Nous avons implémenté cette méthode pour voir si la diagonalisation peut donner de bons résultats avec la méthode vectorielle (c.à.d. l'utilisation de l'image sous forme de vecteur) ; le principe est le même que celui de la Diag(2D)PCA mais nous utilisons la PCA, ainsi les étapes d'apprentissage et de test sont les suivantes :

Apprentissage :

1. Diagonaliser les images d'apprentissage
2. Transformer les images (originales & diagonales) en vecteur 1D et les concaténer dans une seule matrice-apprentissage (originale & diagonale respectivement)
3. Calculer le vecteur moyen diagonal
4. Soustraire la moyenne diagonale aux images d'apprentissage originales et diagonales
5. Calculer la matrice de corrélation diagonale
6. Calculer les valeurs et les vecteurs propres diagonaux
7. Projeter les images d'apprentissage originales avec les vecteurs propres diagonaux.

Test :

1. Transformer l'image-test (ou les images-test) en vecteur et les concaténer dans une seule matrice-test
2. Soustraire la moyenne diagonale au test
3. Projeter l'image-test (ou les images-test) avec les vecteurs propres diagonaux
4. Trouver le voisin le plus proche selon le choix de la distance.

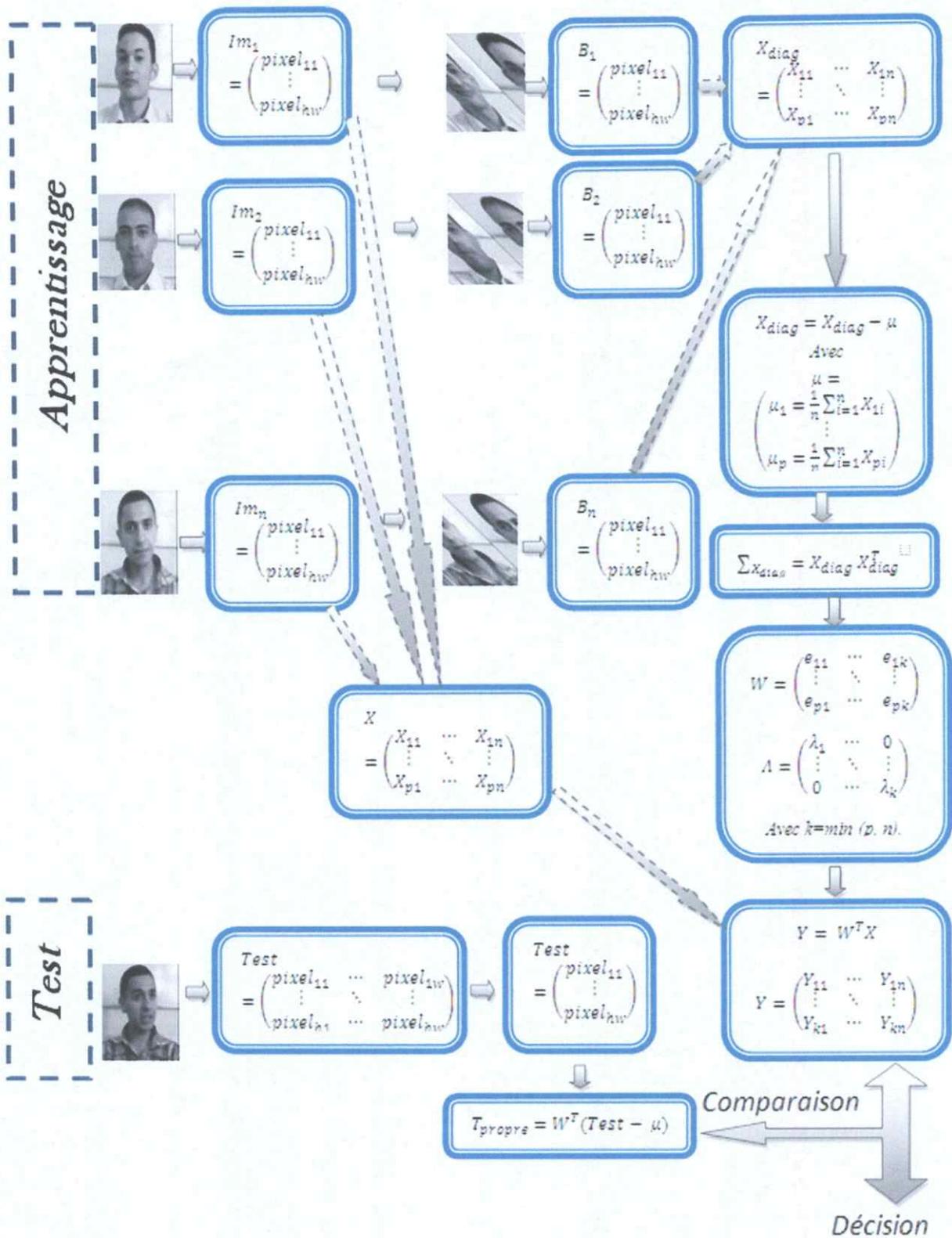


Figure VI.7 : Processus de reconnaissance par la Diagonale PCA

VI.4. Diagonale DCT:

Le principe est de diagonaliser les images-apprentissage et les images-test avant l'application de la DCT et de calculer le voisin le plus proche selon le choix de la distance.

VI.5. Diagonale (2D) PCA dans le domaine DCT:

Dans cette approche, nous avons appliqué la diagonale (2D) PCA dans le domaine DCT. La transformée DCT 2D globale a été utilisée comme étape d'extraction de caractéristiques sur des images aux préalables diagonalisées, un seul bloc de taille $w*w$ (le coin supérieur gauche) contenant les premiers coefficients de la matrice DCT 2D est choisi puisque l'information la plus importante est concentrée dans ces coefficients, ainsi la structure 2D est gardée et la réduction de dimension est réalisée.

Les étapes d'apprentissage et de test sont illustrées par les étapes suivantes :

Apprentissage :

1. Diagonaliser les images-apprentissage
2. Appliquer la DCT Globale sur les images diagonales
3. Appliquer la (2D) PCA sur des images DCT de taille réduite c'est-à-dire des blocs DCT (2D) de dimension $w*w$.

Test :

1. Diagonaliser l'image-test
2. Appliquer la DCT Globale sur l'image-test
3. Projeter l'image-DCT (un bloc de taille $w*w$) avec les vecteurs propres de la diagonal (2D) PCA
4. Calculer le voisin le plus proche selon le choix de la distance.

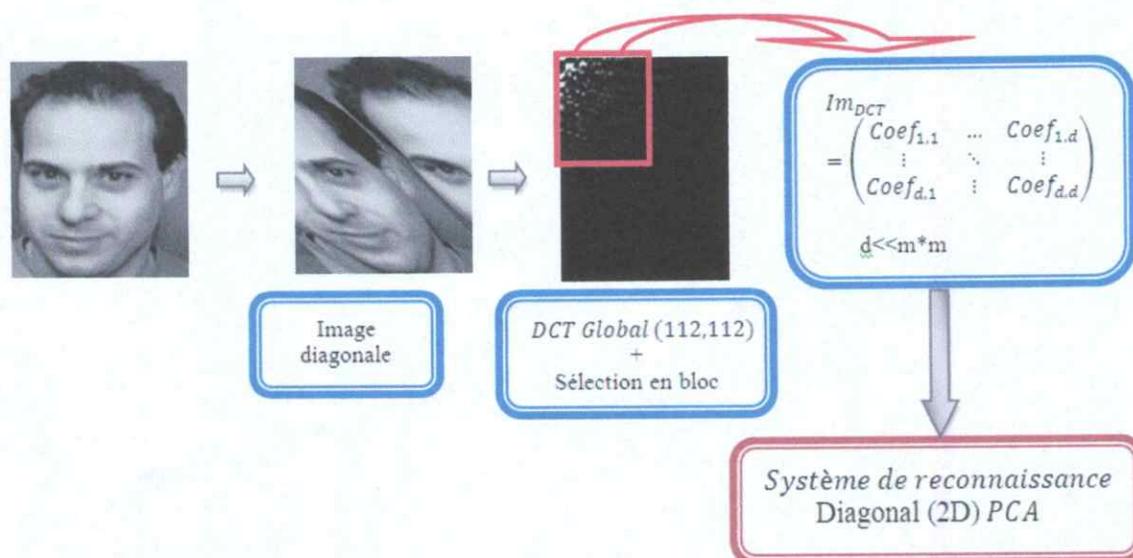


Figure VI.8 : *Processus de reconnaissance par Diagonal (2D) PCA dans le domaine DCT*

VI.6. Diagonale PCA dans le domaine DCT:

Dans cette approche, nous avons appliqué la diagonale PCA dans le domaine DCT. La transformée DCT 2D par bloc a été utilisée comme étape d'extraction de caractéristiques.

Les étapes d'apprentissage et de test sont illustrées par les étapes suivantes :

Apprentissage :

1. Appliquer la DCT Bloc sur les images diagonales (les images DCT sont sous forme de vecteur) et les concaténer dans une seule matrice apprentissage
2. Appliquer la PCA à la matrice-apprentissage.

Test :

1. Appliquer la DCT Bloc sur l'image-test
2. Soustraire la moyenne à l'image-test DCT
3. Projeter l'image-test avec les vecteurs propres
4. Calculer le voisin le plus proche selon le choix de la distance.

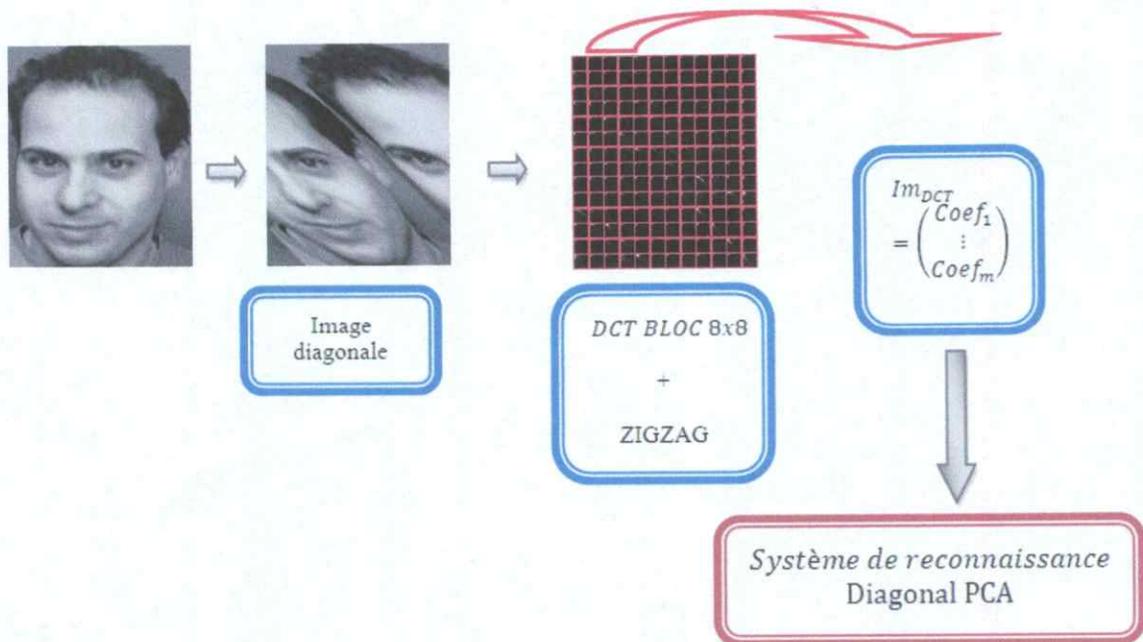


Figure VI.9 : Processus de reconnaissance par Diagonal PCA dans le domaine DCT

Conclusion :

Toutes les méthodes que nous étudions dans notre travail de recherche sont basées sur la décorrélation. Comme il y a une très grande corrélation dans une image, la séparation ne donne pas un bon résultat, la diagonalisation génère une discontinuité dans l'image donc la décorrélation tire plus d'informations des discontinuités, ce qui donne un taux de reconnaissance plus élevé. La partie qui suit sera consacrée à la conception et à la réalisation de l'application.

PARTIE 3 :
Partie pratique

Chapitre VII :
Conception et réalisation

Introduction :

Ce chapitre est consacré à la partie conception et réalisation de notre système de reconnaissance de visage. La méthode utilisée est la méthode UML (Unified Modeling Language). UML est un langage de modélisation des systèmes informatique permet de modéliser, visualiser ou construire un système informatique, elle se base sur le concept objet.

VII.2. Pourquoi UML ?

Cette approche présente plusieurs avantages dont les principaux sont :

- La simplicité de la représentation.
- La facilité de décodage et de réutilisation
- Similitude avec le réel
 - description plus précise des combinaisons (données, opérations)
 - décomposition basée sur “classification naturelle”
- La stabilité de la conception en cas de modification du système ou de maintenance.

VII.3. Démarche utilisée :

On a choisi la démarche RUP (Rational Unified Process), cette démarche utilise les diagrammes suivants :

- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme de classe
- Diagramme de collaboration

VII.4. Conception :

La modélisation consiste à représenter un système par un ensemble de diagrammes d'utilisation expliquant le fonctionnement du système par l'illustration des différents cas d'utilisation, ainsi que les acteurs qui interagissent avec le système, et un diagramme de classe qui montre la coopération entre les différentes classes du système pour qu'il soit opérationnel.

VII.4.1. Les acteurs :

Un acteur représente un rôle joué par une personne ou autre,

Dans notre projet nous avons deux acteurs, chaque acteur a un rôle bien précis :

1. Administrateur : C'est la personne qui gère le système à travers une interface.
2. La Personne test : C'est la personne que le système va le tester.

VII.4.2. Les cas d'utilisations :

Ils permettent de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système ! Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système.

1. Pour l'administrateur :

- Gérer la base de données (ajout/suppression/modification) d'une personne.
- Capturer une personne.
- Faire un apprentissage avec le choix d'une meilleure configuration et une meilleure approche pour assurer l'efficacité du système.
- Faire un test que soit d'une personne test ou d'une base de données pour avoir le taux de reconnaissance.

2. Pour la personne test :

- Se positionner devant un capteur et attendre une réponse.

VII.4.3. Diagramme de cas d'utilisation :

Un diagramme de cas d'utilisation permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c.-à-d. toutes les fonctionnalités que doit fournir le système.

Ils sont utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Les deux composants principaux des diagrammes de cas d'utilisation sont les acteurs et les cas d'utilisation.

Pour l'administrateur :

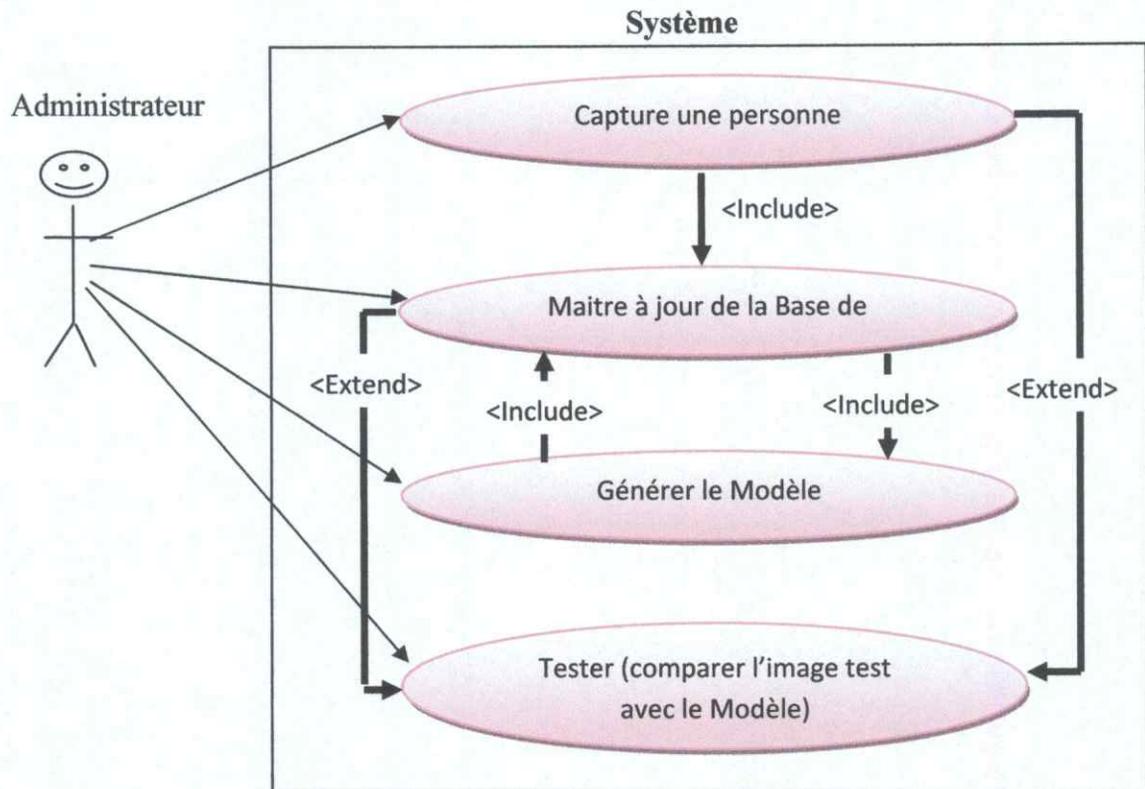


Figure VII.1 : Diagramme de cas d'utilisation pour l'administrateur.

VII.4.4. Diagramme de séquence :

Un diagramme de séquence est une représentation du déroulement des traitements et de l'interaction entre les éléments du système et de ses acteurs, il décrit une séquence de message entre des classes.

Pour l'administrateur :

- **Gestion de base de données :**

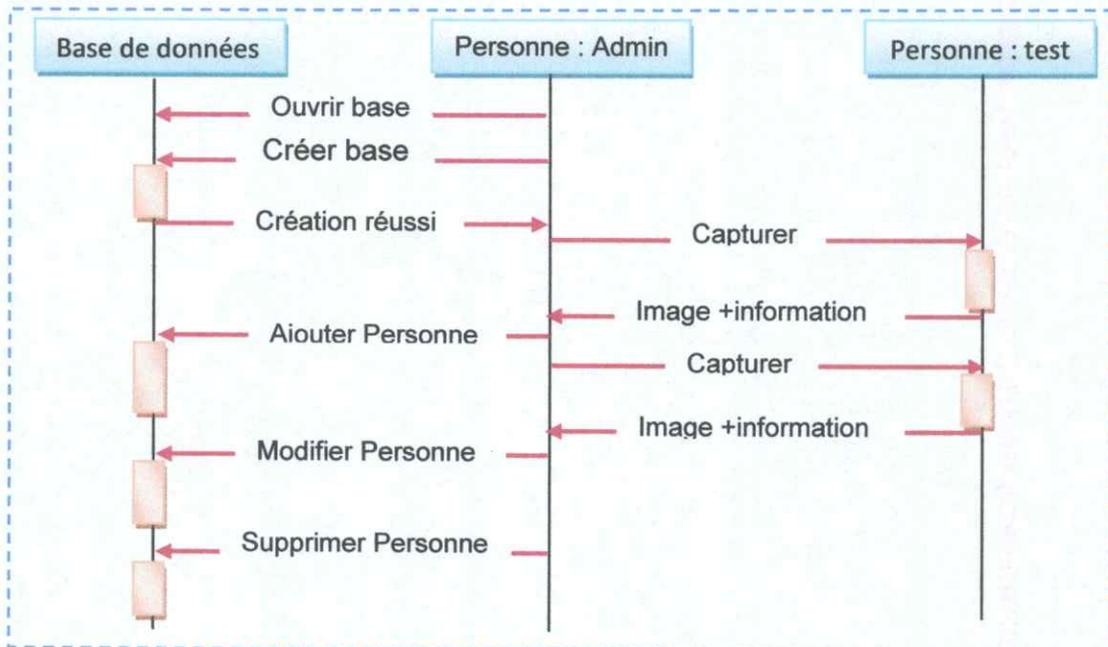


Figure VII.2 : Diagramme de séquence pour la gestion de base de données.

Dans la création d'une base de données, le système crée toutes les tables de la base de données qui sont :

- Personne
- Visage
- Configuration
- Modèle de chaque approche
- Taux

• **Apprentissage :**

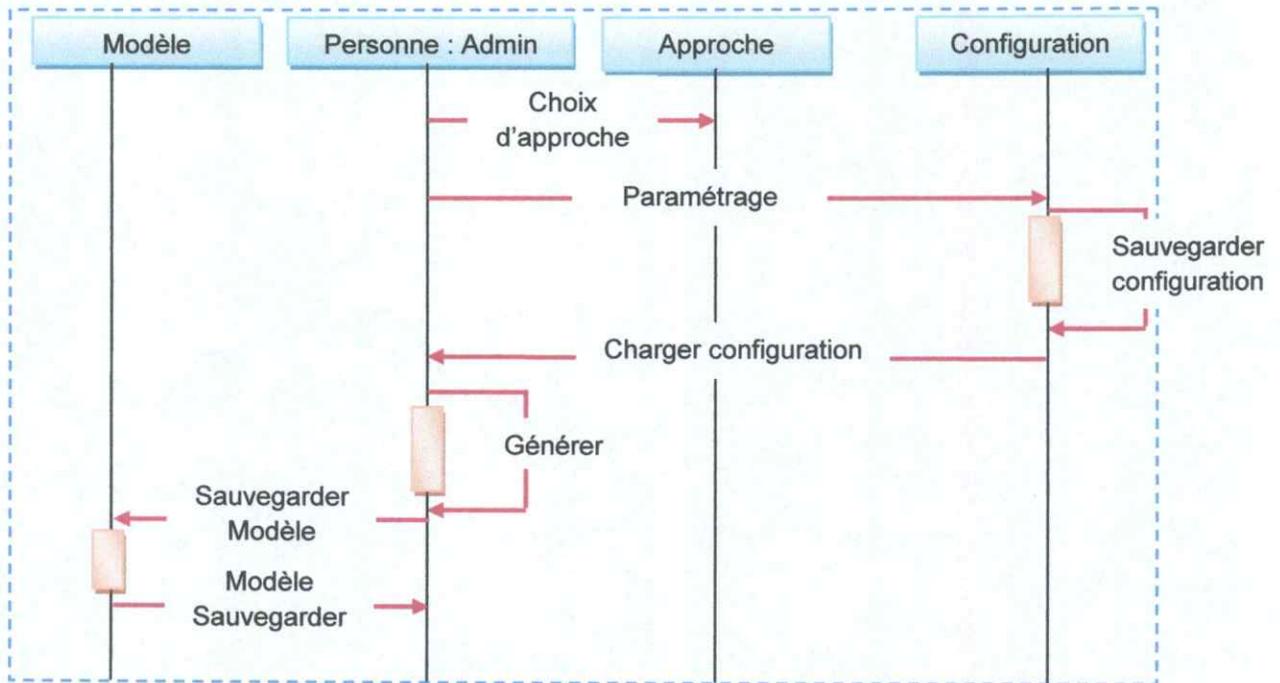


Figure VII.3 : Diagramme de séquence pour l'apprentissage.

• **Test :**

• pour une base de données :

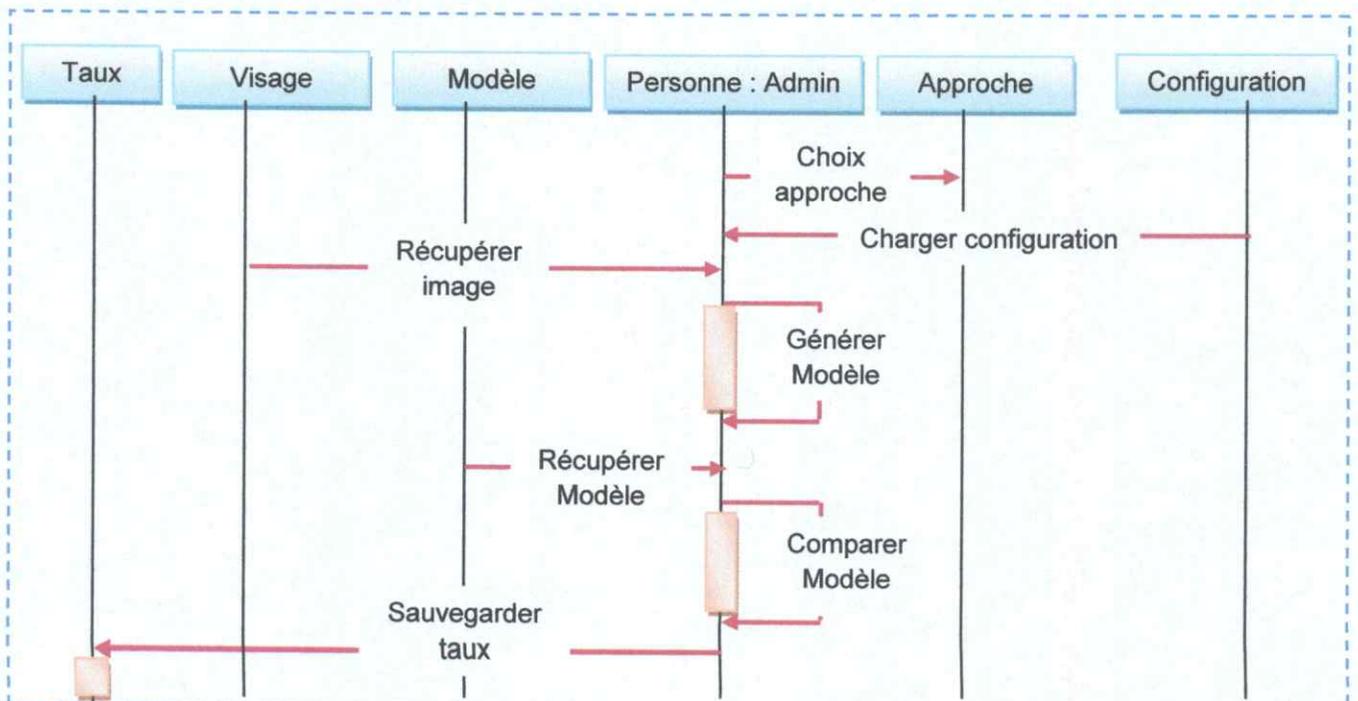


Figure VII.4 : Diagramme de séquence pour un test global.

- pour une personne test :

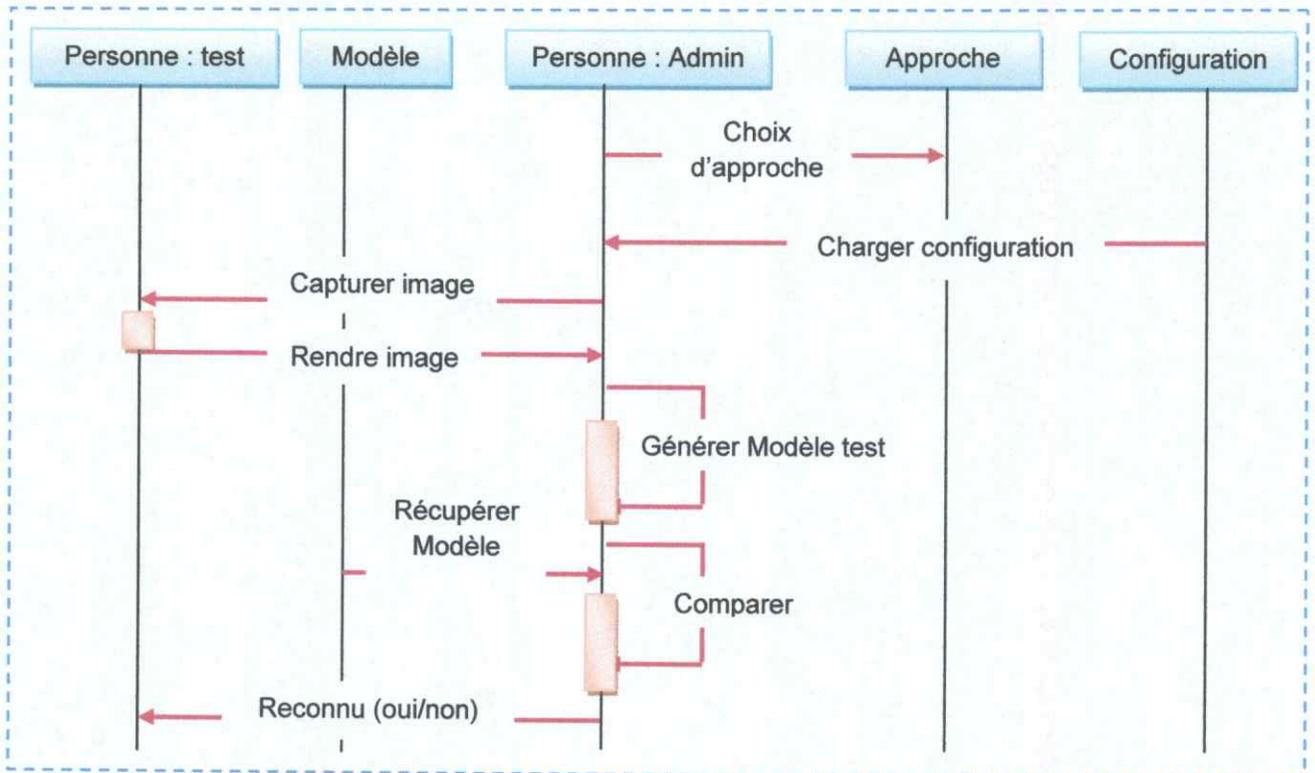


Figure VII.5: Diagramme de séquence pour un test d'un Person.

VII.4.5. Diagramme de classe :

Le diagramme de classe représente la structure du système sous forme de classe et de relation entre ces classes.

Pour mieux simplifier notre modélisation et la structure statique de notre modèle nous avons devisé notre représentation en plusieurs modules selon la relation et l'interaction entre les classes.

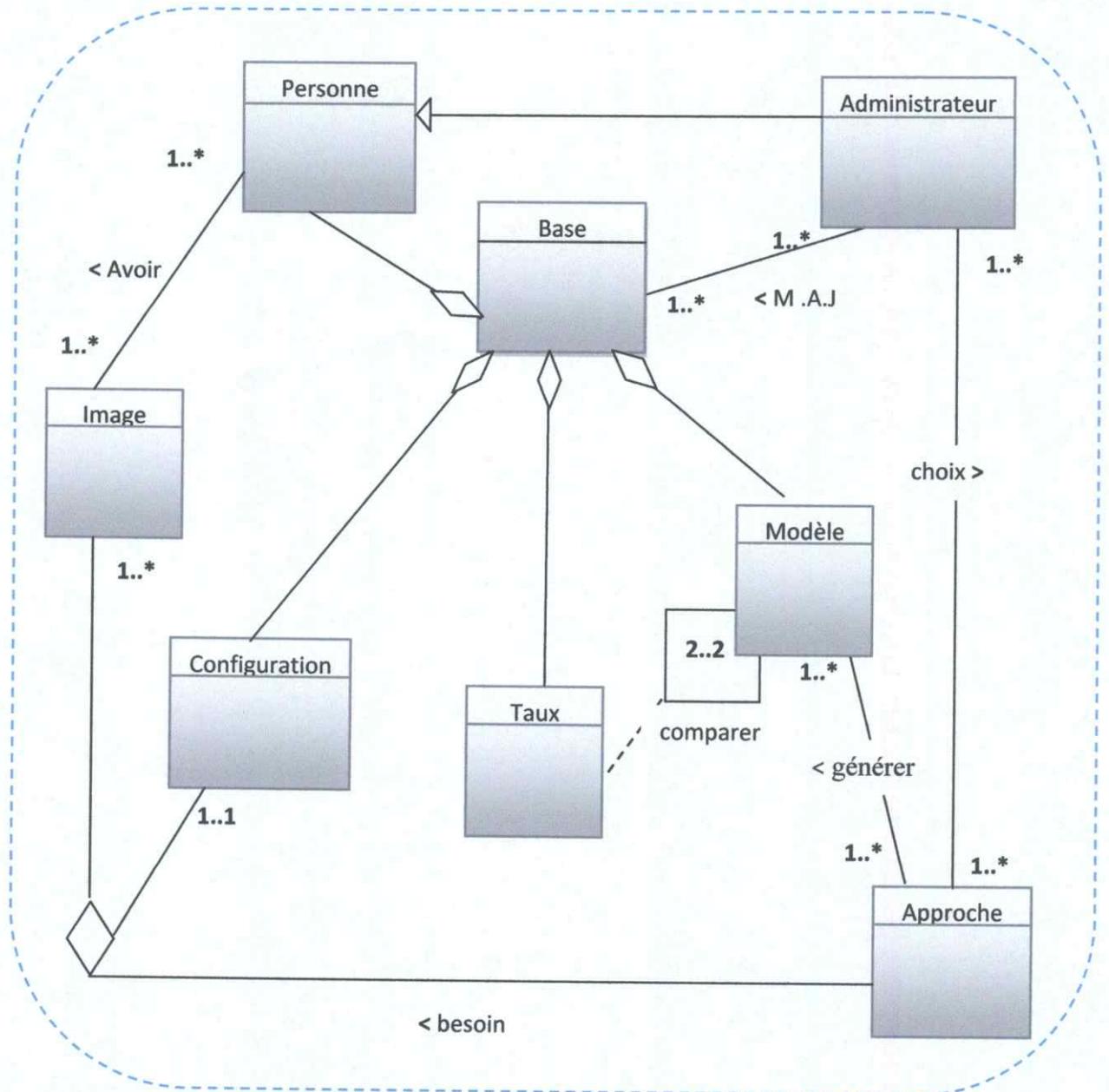


Figure VII.6 : Diagramme de classe pour la gestion d'une base de données.

Description des classes :**Classe Personne :**

Attribut	Type	signification
id_person	int	Identifiant de la personne
nom	Varchar[50]	Nom de la personne
prenom	Varchar[50]	Prénom de la personne
Fonction	Varchar[50]	La fonction de la personne
Date_N	Date	La date de naissance de la personne

Table VII.1. La classe personne**Les méthodes de la classe personne :**

Méthode	signification
Get_person()	Retourner le nombre de personne
Ajouter_person()	Ajouter une nouvelle personne
Modifier_person()	Modifier une personne
Supprimer_person()	Supprimer une personne
Verifier_existe()	Vérifier l'existence de la personne

Table VII.2. Les méthodes de la classe personne**Classe image :**

Attribut	Type	Signification
Id_image	Int	Identifiant de l'image
Id_personne	Int	Identifiant de la personne
Im_pixel	image	Image de la personne

Table VII.3. La classe image

Les méthodes de la classe image:

Méthode	Signification
Get_nbr_image(base)	Retourner le nombre d'image dans la base de données
Ajouter_image()	Ajouter une nouvelle image
Modifier_image()	Modifier une image
Supprimer_image()	Supprimer une image
Get_image(int)	Retourner l'image de l'indice int

Table VII.4. Les méthodes de la classe image**Classe Configuration :**

Attribut	Type	Signification
ID_Config	Int	Identifiant de la configuration
Nbr_pers	Int	Nombre de personne
Pose	Int	Nombre de pose par personne
Coeff_dct	Int	Nombre de coefficient dct
chauv	Int	Chevauchement entre bloc dct
Taille_bloc	Int	Nombre de bloc de dct global
Comp_pca	Int	Nombre de composant pca
Col_2d	Int	Nombre de colonne de 2dpca

Table VII.5. La classe configuration**Les méthodes de la classe Configuration :**

Méthode	Signification
Ajouter_config()	Ajouter une nouvelle configuration

Table VII.6. Les méthodes de la classe configuration

Classe modèle :

Attribut	Type	Signification
ID_C	Int	Identifiant de la configuration
Modèle	binary	Modèle de l'apprentissage

Table VII.7. La classe modèle

Dans notre cas, la classe modèle est une classe mère elle se divise en plusieurs sous classes et qui sont :

Modèle_dct, Modèle_dctg, Modèle_pca, Modèle_dpca, Modèle_2dpca,
Modèle_d2dpca, Modèle_dctpca, Modèle_dctdpca, Modèle_dctg2dpca,
Modèle_dctgd2dpca.

Elles sont classées selon 2 catégories :

Categorie1 :

Modèle_dct, Modèle_dctg.

Categorie2:

Modèle_pca, Modèle_dpca, Modèle_2dpca, Modèle_d2dpca, Modèle_dctpca,
Modèle_dctdpca, Modèle_dctg2dpca, Modèle_dctgd2dpca.

Format de la classe catégorie 1:

Attribut	Type	Signification
ID_C	Int	Identifiant de la configuration
Modèle	Binary	Modèle de l'apprentissage

Table VII.8. Format de la classe catégorie 1

Format du classe catégorie 2 :

Attribut	Type	Signification
ID_C	Int	Identifiant de la configuration
Modèle	Binary	Modèle de l'apprentissage
Modèle_m	Binary	Modèle de la moyenne d'images
Modèle_Vect	Binary	Modèle des vecteurs propres

Table VII.9. Format de la classe catégorie 2**Les méthodes de la classe modèle:**

Méthode	Signification
Save_Modèle()	Sauvegarder un modèle
Verifier_existe()	Vérifier si le modèle existe dans la BDD

Table VII.10. Les méthodes de la classe Modèle**Classe approche :**

Attribut	Type	Signification
Nom_app	Varchar [50]	Nom de l'approche
param	Int	Paramètre de l'approche

Table VII.11. Format de la classe approche

Dans notre cas, la classe approche est une classe mère elle se divise en plusieurs sous classe et qui sont :

DCT, DCTGLOB, PCA, DPCA, (2D)PCA, D(2D)PCA, DCT_PCA, DCT_DPCA, DCTG_(2D)PCA, DCTG_D(2D)PCA.

Elles sont classes selon 3 catégories :

Catégorie 1: avec 2 paramètres

DCT, DCTG_(2D)PCA, DCTG_D(2D)PCA.

Categorie 2: avec 1 paramètre

DCTGLOB, PCA, DPCA, (2D)PCA, D(2D)PCA.

Categorie3: avec 3 paramètres

DCT_PCA, DCT_DPCA.

Les classes Modèle et approche sont les mêmes que les précédents.

Classe taux :

Attribut	Type	Signification
ID_C	Int	Identifiant de la configuration
Approche	Varchar[50]	Nom de l'approche
Taux_rec	Double	Taux de reconnaissance

Table VII.12. Format de la classe taux

Les méthodes de la classe taux :

Méthode	Signification
Save_taux()	Sauvegarder le taux de reconnaissance
Calcule_taux()	La fonction qui calcule le taux de reconnaissance
Normes()	Calculer la distance entre deux individus
Norme_test()	Calculer la distance entre deux individus

Table VII.13. Les méthodes de la classe taux

VII.4.6. Diagramme de collaboration :

Le diagramme de collaboration décrit le comportement collectif d'un ensemble d'objet. En vue de réaliser une opération en décrivant leurs interactions modélisées par des envois numérotés de messages.

• **Gestion de base de données :**

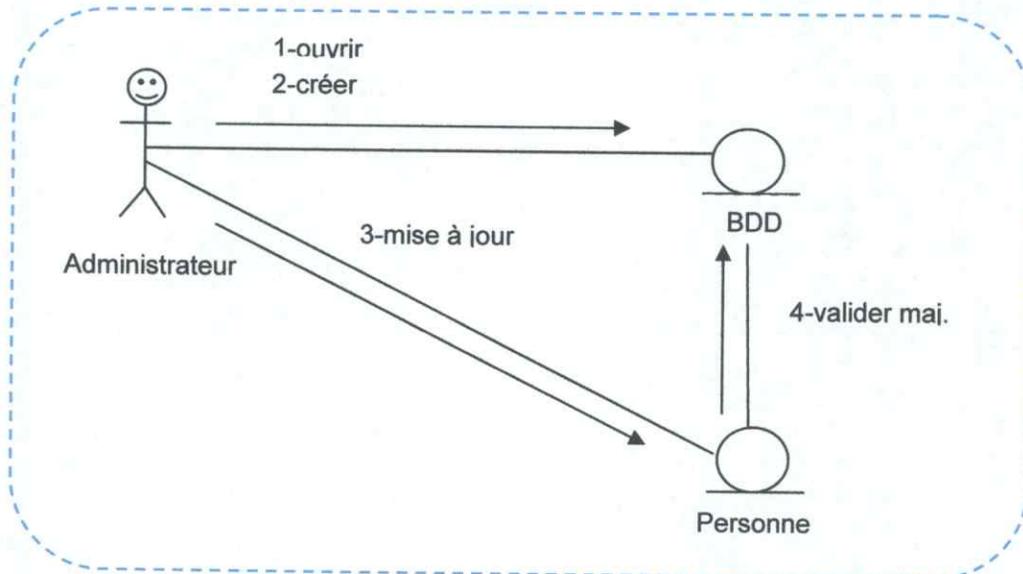


Figure VII.7 : Diagramme de collaboration pour la gestion de BDD

• **Apprentissage**

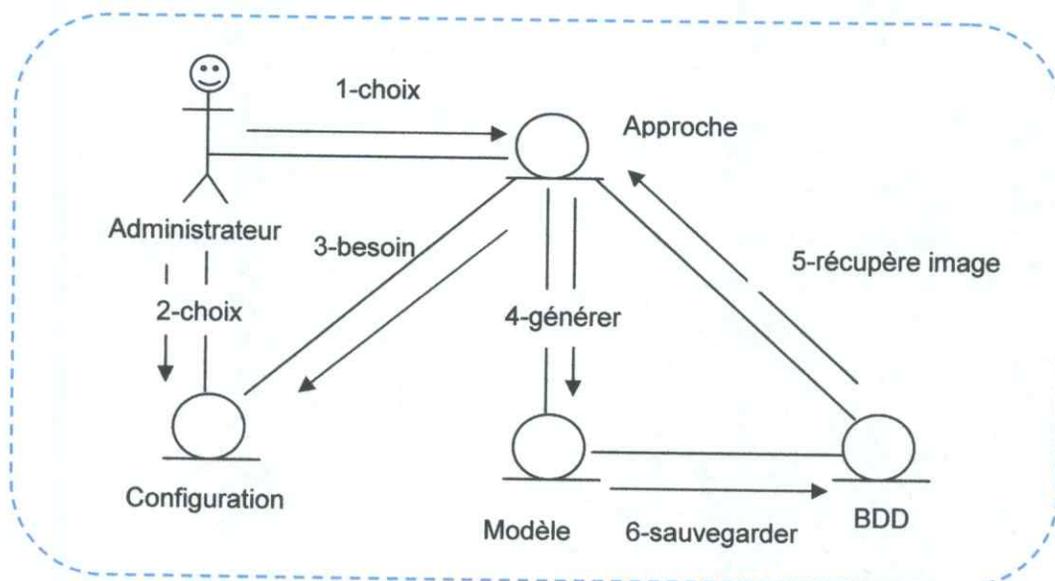


Figure VII.8 : Diagramme de collaboration pour l'apprentissage.

• Test :

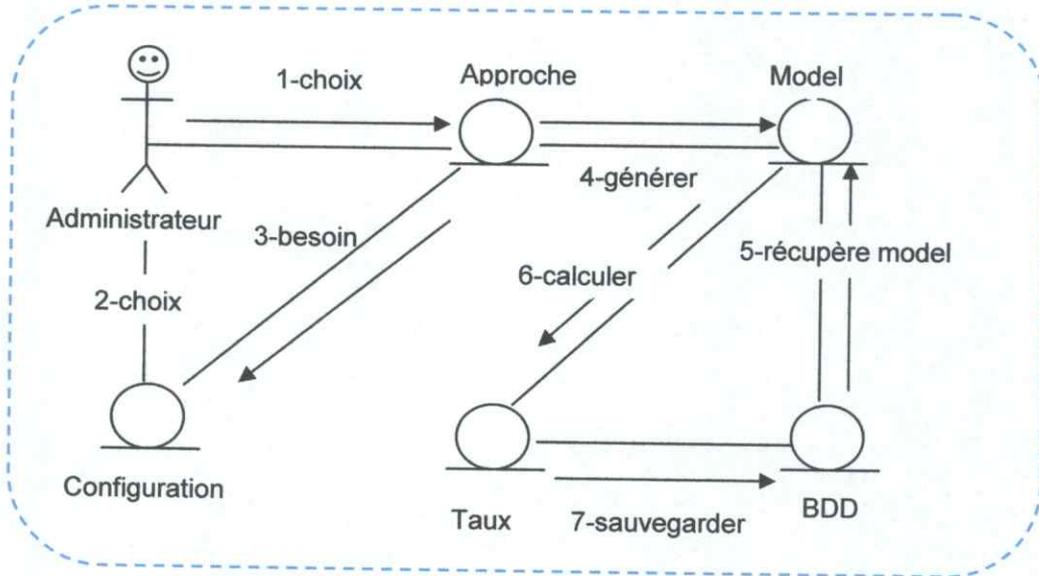


Figure VII.9 : Diagramme de collaboration pour un test.

• Test pour une personne test:

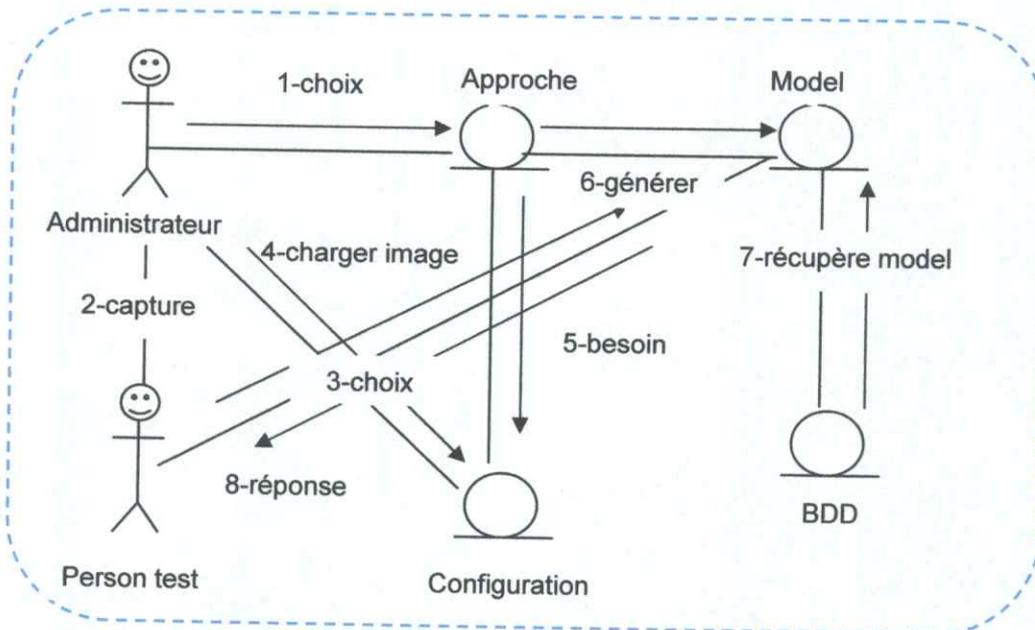


Figure VII.10 : Diagramme de collaboration pour une personne test.

VII.5. Réalisation :

VII.5.1. Introduction :

Dans cette partie, nous allons décrire les moyens utilisés suivies pour la réalisation de notre système.

VII.5.2. Les outils de développement :

On a utilisée deux outils qu'utilise le CDTA pour développer notre système, le premier c'est Visual C++.NET pour l'application et le deuxième SQL Server pour gérer les bases de données des personnes et leurs visages et les modèles d'apprentissage.

VII.5.2.1. Microsoft Visual C++.NET :

Visual C++ .NET est un ensemble d'outils performants de Microsoft, destiné à créer des applications Windows de haut niveau ainsi que des applications et des composants .NET à l'aide du langage de développement C++. Cet environnement de développement robuste permet la modification et le débogage efficace du code source.

On cite parmi les possibilités qu'il offre:

- Créer des applications et des composants .NET d'une grande précision.
- Utiliser les extensions managées destinées à C++ ainsi que la prise en charge du compilateur pour une génération de code MSIL (Microsoft Intermediate Language) optimisée.
- Le code C++ existant peut être compilé pour l'environnement .NET sans qu'il soit nécessaire de le réécrire. Visual C++ prend en charge la combinaison de codes et de données managés et non managés.
- Utiliser des bibliothèques pour incorporer des fonctionnalités avancées, Visual C++ .NET inclut une variété de bibliothèques contenant de nombreux types de données et modèles de programmation.

La plate forme .NET se base sur le framework.NET qui fournit des bibliothèques de classes très variées et structurées, et donc il fournit un véritable environnement objet en plus il permet l'interopérabilité des langages de programmation (exemple Visual Basic, J# , C++,C#...etc.).

VII.5.2.2. SQL SERVER 2000 :

SQL server est un SGBDR (Système de Gestion de Base de Donnée Relationnelle) de Microsoft.

On cite parmi ses avantages :

- Administration aisée.
- Indépendance entre les diverses bases, facilitant l'intégration de plusieurs applicatifs dans une même instance.
- Une des bases les plus performantes sous Windows en configuration par défaut.
- Optimiseur statistique enrichi à flux tendu.
- Réplication intégrée.
- Langage T-SQL très convivial.
- Procédures stockées simples à écrire.
- Sous-SELECT possible dans clause FROM.
- Services Web.
- Support XML.

Ce qui fait de SQL Server un SGBD complet et fiable et donc idéal pour le développement rapide d'applications performantes, en plus il est très adapté avec Visual Studio.NET.

Ce sont les outils logiciels que nous avons utilisés pour l'élaboration de notre système; vous remarquez que c'est une plate-forme cent pour cent Microsoft, ce qui offre une certaine homogénéité et compatibilité.

VII.5.3. Architecture de notre système de reconnaissance de visage :

Notre système est composé de plusieurs modules et qui sont :

- Acquisition
- Prétraitement
- Apprentissage
- Test

Et voici un organigramme qui montre le fonctionnement du système :

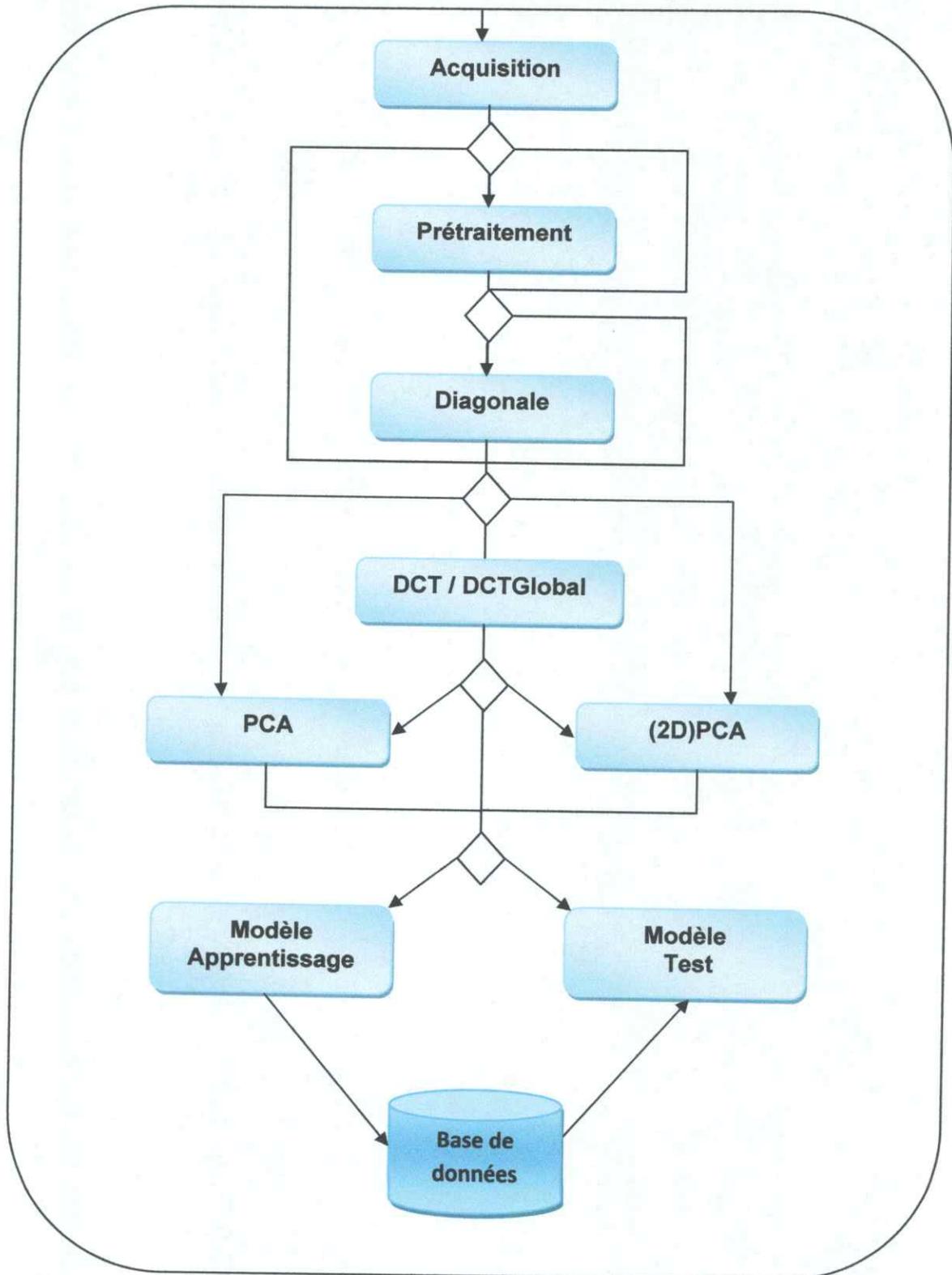


Figure VII.11 : Schéma conceptuel du système.

VII.5.4. Les modules de l'application :

On présentera les différentes interfaces de l'application, chacune d'elles correspond à un module du système.

Commençons par la fenêtre principale de l'application :

VII.5.4.1. Forme principale :

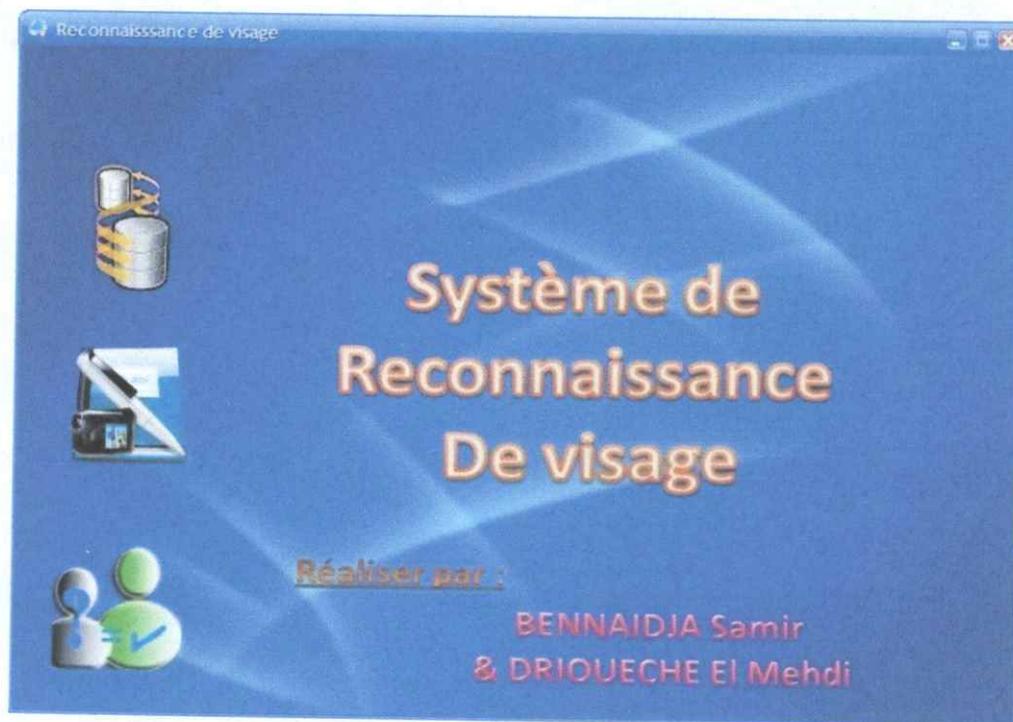


Figure VII.12 : La forme principale du système.

A partir de cette forme, on peut accéder à la base de donnée à travers la première image, l'apprentissage à travers la 2^{ème} image et le test à travers la 3^{ème} image.

VII.5.4.2. Base de données :

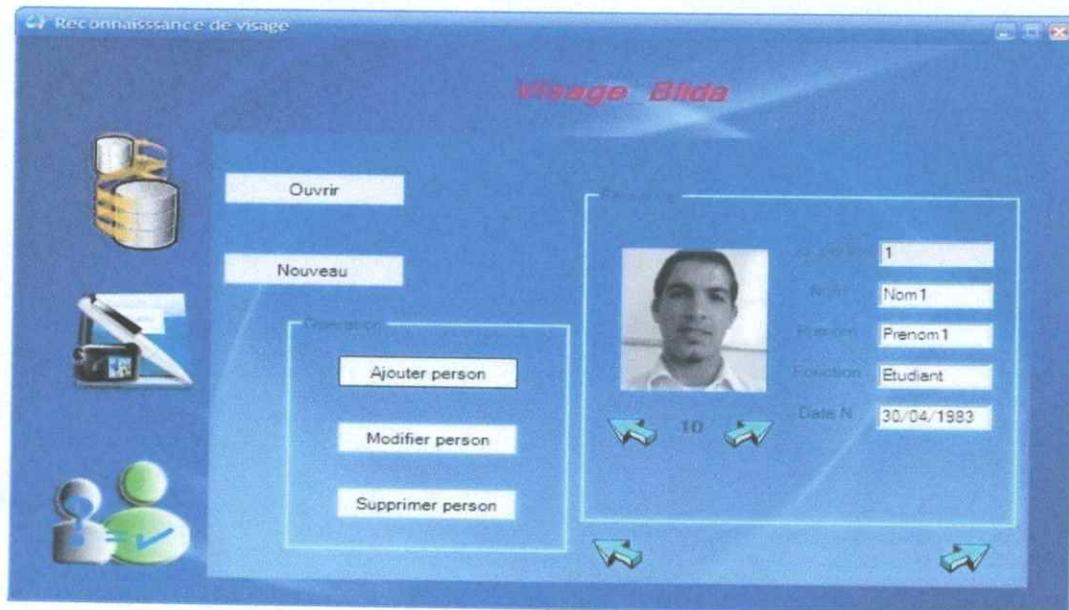


Figure VII.13 : La forme de base de données.

A travers cette fenêtre, on peut créer ou ouvrir une base de données et aussi ajouter, modifier ou supprimer les données d'une Personne que soit image ou coordonnée.

VII.5.4.3. Apprentissage :

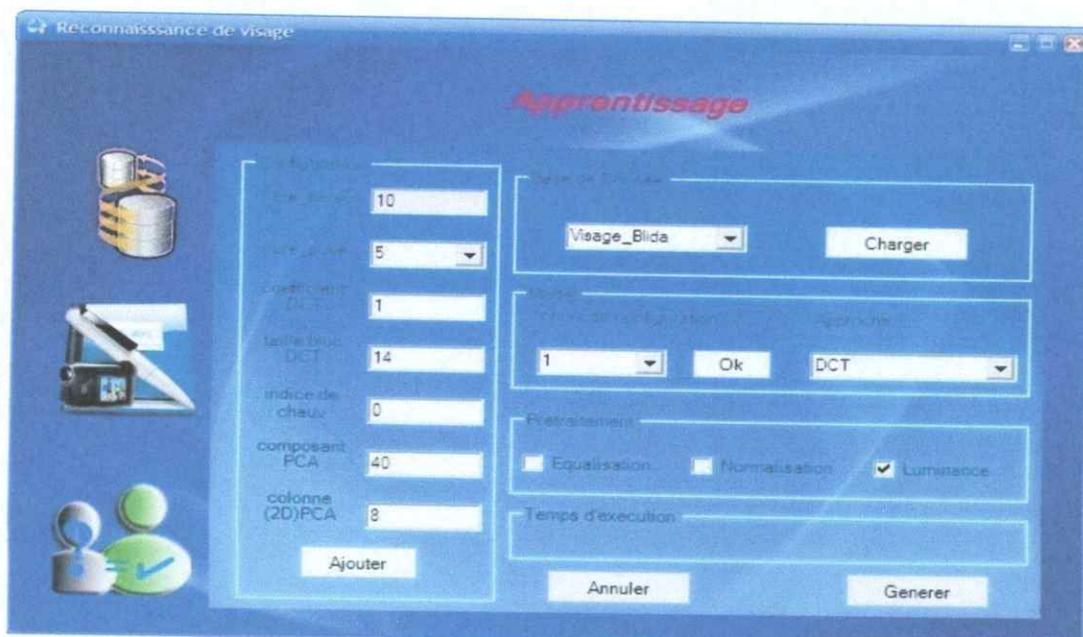


Figure VII.14 : Forme de l'apprentissage.



A travers cette fenêtré (apprentissage), on peut :

- Ajouter ou charger une configuration.
- Générer le modèle de l'apprentissage.
- Sauvegarder le modèle dans la base de données.

VII.5.4.4. Test :



Figure VII.15 : Forme de test.

A travers cette forme, on peut :

- Tester une personne test et voir le résultat.
- Faire un test global de la base de données.
- Calculer le taux de reconnaissance et les sauvegarder dans la base de données.
- Tracer le graphe de taux de reconnaissance pour chaque approche.

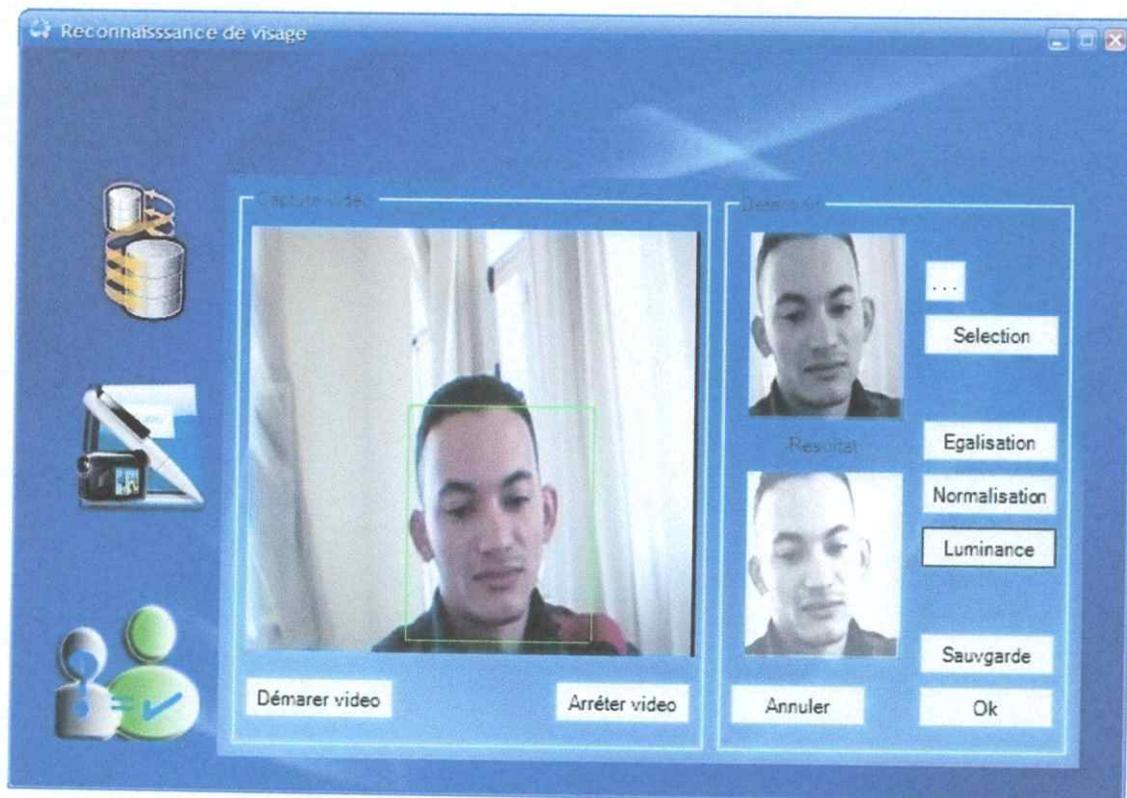
VII.5.4.5. Acquisition et prétraitement :

Figure VII.16 : Forme d'acquisition.

A travers cette forme on peut :

- Capturer une personne avec la détection de visage.
- Appliquer des prétraitements à une image et la sauvegarder.
- Les prétraitements appliqués sur une image sont :
 - Egalisation.
 - Normalisation.
 - Luminance.

Conclusion :

À travers ce chapitre, nous avons vu les différentes étapes de la conception et de la réalisation du système.

Pour la conception, nous avons défini les acteurs du système et leurs rôles, les cas d'utilisation pour chaque acteur et les différents diagrammes : diagramme de cas d'utilisation, diagramme de séquence, diagramme de classe et enfin le diagramme de collaboration.

Pour la réalisation, nous avons défini un schéma conceptuel du système, nous avons vu ses différents modules et le fonctionnement de chaque module.

Il nous reste, maintenant, à l'évaluer par des tests, ce que nous développerons dans le chapitre suivant et qui constituera la partie finale de notre travail.

Chapitre VIII :
Etude comparative

Introduction :

Dans ce chapitre, nous citerons toutes les approches vues dans notre application et illustrerons chacune d'elles par un graphe qui mettra en évidence les performances de chacune et ce, afin de pouvoir comparer les deux approches DCT(*Discrete Cosines transform*)-PCA(*Principal composant analysis*) et DCT-DPCA (*Diagonal Principal composant analysis*)et dégager laquelle est la plus performante.

VIII.1. Base de données utilisée:VIII.1.1. ORL :

Conçue par AT&T Laboratoires de l'université de Cambridge en Angleterre, l'ORL¹ (Olivetti Research Laboratory) est une base de visages de référence pour les systèmes de reconnaissance automatiques de visages.

Elle est constituée de 400 images de 40 individus. Pour chaque individu, on dispose de 10 images prises à des intervalles de temps différents avec des variations en expression faciale (les yeux ouverts/fermés, changement d'inclinaison, ...), variations des détails du visage (coupes de cheveux, lunettes) et de faibles variations en luminance. Toutes les images sont prises avec un arrière-plan homogène noir. La taille des images est 112x92 pixels où chaque pixel est codé sur 256 niveaux de gris. Nous présentons ci-après deux figures illustrant les visages de cette base et leurs variations :



Figure VIII.1 : Variation en inclinaison, échelle et expressions faciales de la base ORL.

¹ La base ORL est accessible sur le lien : <http://www.cam-orl.co.uk/facedatabase.html>



Figure VIII.2: Variation en coupe de cheveux, barbes, lunettes et luminance de la base ORL.

VIII.2.2. Blida Visage :

Cette base de données est une base réelle des étudiants en informatique de l'université SAAD DAHLEB de Blida.

Elle est constituée de 100 images de 10 individus, pour chaque individu, nous avons disposé de 10 images prises en même temps avec des variations en expression faciale (les yeux ouverts/fermés, changement d'inclinaison, ...), variations des détails du visage (lunettes) et de faibles variations en luminance. Toutes les images ont été prises avec un arrière-plan homogène. La taille des images est 112x92 pixels où chaque pixel est codé sur 256 niveaux de gris.

Nous présentons ci-dessous deux figures illustrant les visages de cette base et leurs variations :



Figure VIII.3 : Variation en inclinaison, échelle et expressions faciales de la base Blida



Figure VIII.4: Variation en lunettes et luminance de la base Blida

Remarque : Les tests ont été effectués avec la base de données ORL, sous MATLAB 6.5, nous avons pris 5 poses pour l'apprentissage et 5 poses pour le test. Pour le calcul de la distance, nous avons utilisé la norme L1.

VIII.3. Reconnaissance par PCA :

Dans cette partie, on donnera les taux de reconnaissance obtenus par notre système et on comparera entre les différentes méthodes et ceci dans la base ORL.

Les méthodes qu'on va comparer sont :

- ✓ PCA
- ✓ Diagonal PCA
- ✓ (2D) PCA
- ✓ Diagonal (2D) PCA

Les paramètres à évaluer sont les suivants :

- Norme de distance utilisée : L_1 ou L_2 .
- Robustesse par rapport à la luminance : dans ce cas, on générera pour les images tests une luminance artificielle de gauche et de l'ombre à droite suivant cette formule :

$$Im_{lum}(x,y) = Im(x,y) + m \cdot x + \delta$$

avec

$Im_{lum}(x,y)$ c'est le pixel avec de l'image avec luminance et $Im(x,y)$ la valeur du pixel de l'image originale, $x=1...h$, $y=1...w$, $m = \frac{-\delta}{(h-1)}$ et δ c'est le delta d'illumination par pixel et avec h et w longueur et largeur de l'image.



Figure VIII.5 : Application de luminance 60 sur une image

VIII.3.1. PCA :

La PCA est une référence dans le domaine de la reconnaissance de visage et elle ne cesse d'évoluer depuis son application. Ainsi, elle est passée de la PCA à (2D) PCA et récemment à Diagonal (2D) PCA.

VIII.3.1.1. Vecteurs PCA et taux de reconnaissance :

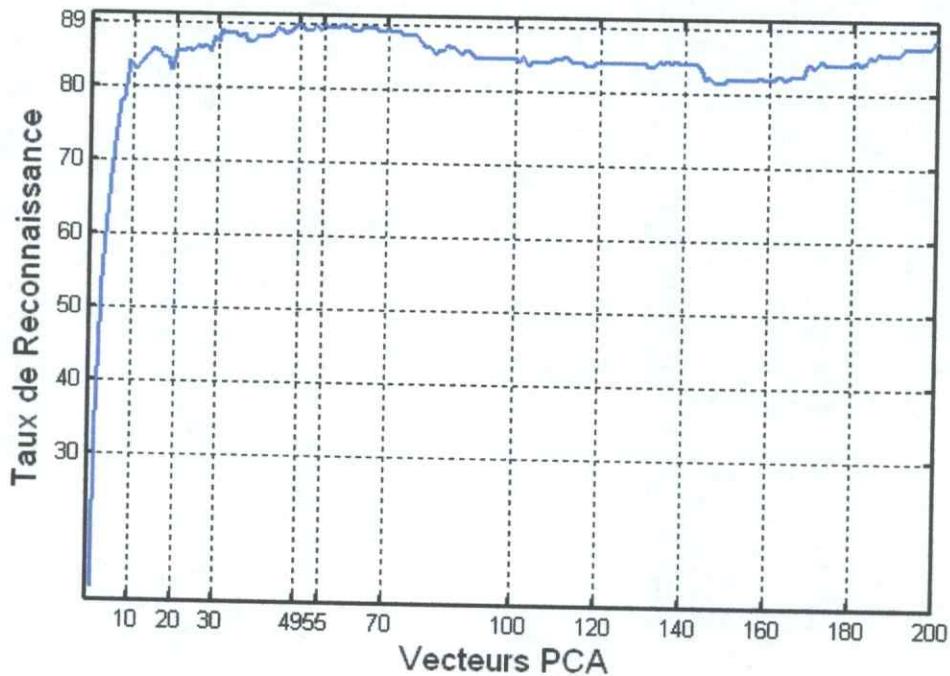


Figure VIII.6: Taux de reconnaissance en fonction du nombre de vecteurs PCA.

Ce graphe nous montre que de 1 à 10 vecteurs, le taux de reconnaissance augmente de manière considérable, cela caractérise la PCA telle que définie au chapitre V. Quand le nombre de vecteurs est de 49 ; 53 ou 55, le taux de reconnaissance atteint son maximum, à savoir 89%.

VIII.3.1.2. avec Luminance :

Sur la base de la remarque qui précède, on a fixé le nombre de vecteurs à 49 et on a varié la luminance de 0 à 80. Nous illustrons l'effet produit sur le taux de reconnaissance dans le graphe qui suit :

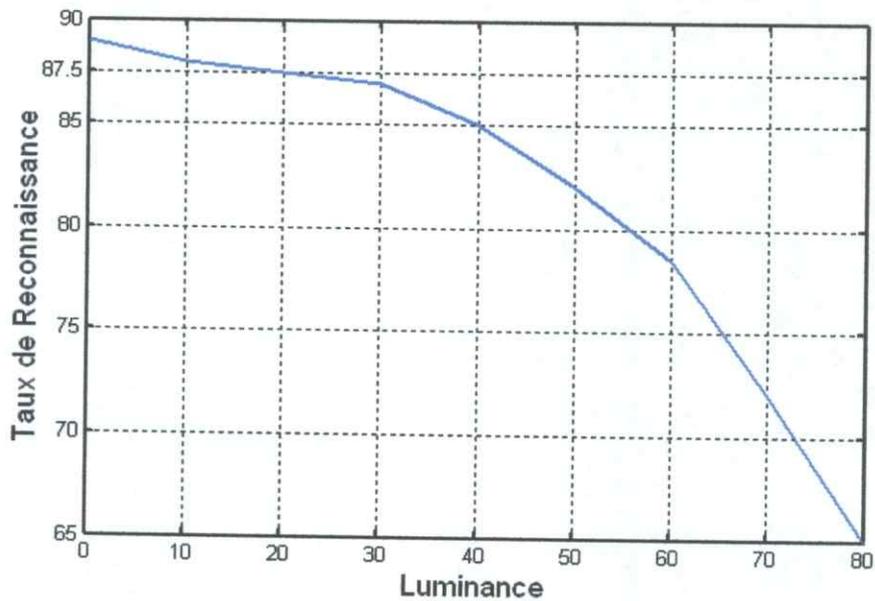


Figure VIII.7 : Taux de reconnaissance en fonction de la luminance

Nous constatons que la luminance altère considérablement le taux de reconnaissance. Nous pouvons affirmer que la PCA est sensible à la luminance.

VII.3.2. DiaPCA (Diagonal PCA) :

En appliquant la diagonale sur la PCA, nous obtenons les résultats suivants :

VII.3.2.1. Vecteurs DiaPCA et taux de reconnaissance :

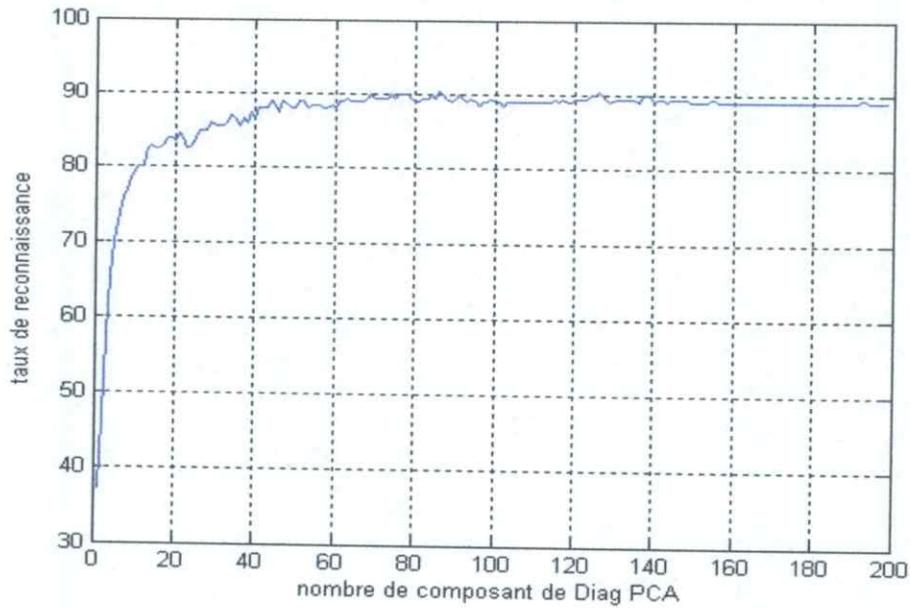


Figure VIII.8: Taux de reconnaissance en fonction du nombre de vecteurs DiaPCA.

Nous obtenons la même courbe qu'avec la PCA mais avec des taux de reconnaissance plus élevés: 90.5% avec 87 vecteurs.

VIII.3.2.2. Luminance :

Sur la base de la remarque qui précède, on fixe le nombre de vecteurs à 87 et on varie la luminance de 0 à 80. Nous illustrons l'effet produit sur le taux de reconnaissance dans le graphe qui suit :

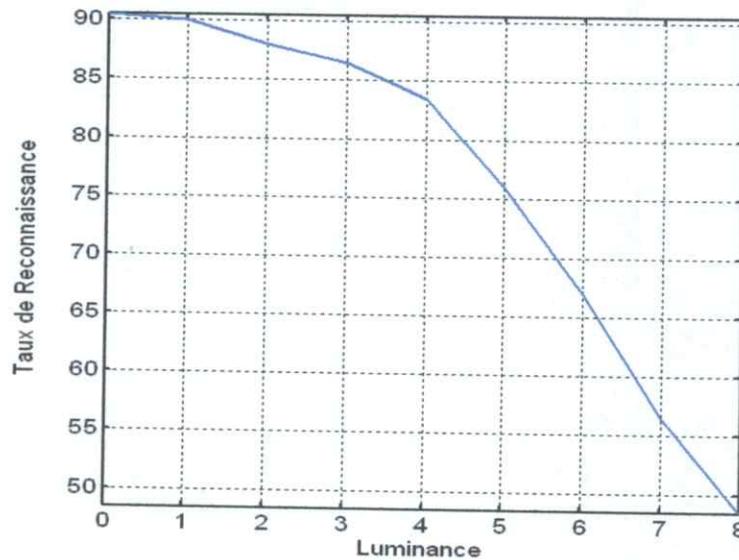


Figure VIII.9: Taux de reconnaissance en fonction de la luminance.

De même que sur la figure VIII.6, nous constatons que la luminance altère considérablement le taux de reconnaissance. Nous pouvons affirmer que la DiaPCA est sensible à la luminance.

VIII.3.3. (2D)PCA :

VIII.3.3.1. Vecteurs (2D)PCA et taux de reconnaissance:

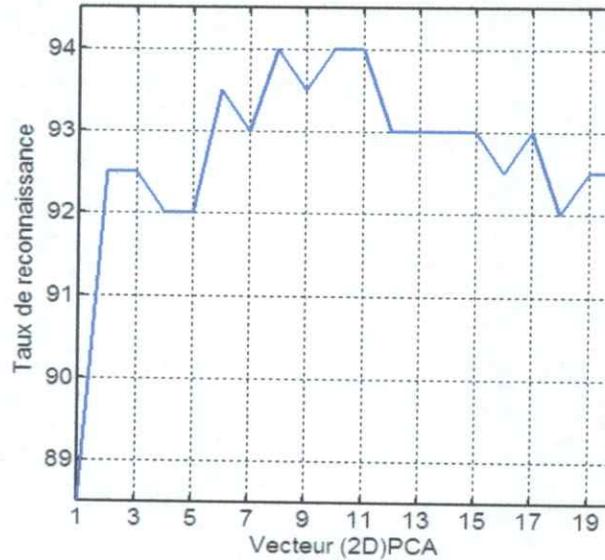


Figure VIII.10: Taux de reconnaissance en fonction du nombre de vecteurs (2D)PCA.

Nous constatons qu'avec un seul vecteur, nous obtenons un taux de reconnaissance de 87%, nous pouvons alors affirmer qu'un seul vecteur permet une bonne discrimination des visages. La (2D)PCA atteint son meilleur taux de reconnaissance quand le nombre de vecteurs atteint 8 ; 10 et 11, à savoir 94% mais à partir de 12, le taux de reconnaissance va en diminuant.

VIII.3.3.2. Luminance :

Sur la base de la remarque qui précède, on a fixé le nombre de vecteurs à 8 et on a varié la luminance de 0 à 80. Nous illustrons l'effet produit sur le taux de reconnaissance dans le graphe qui suit :

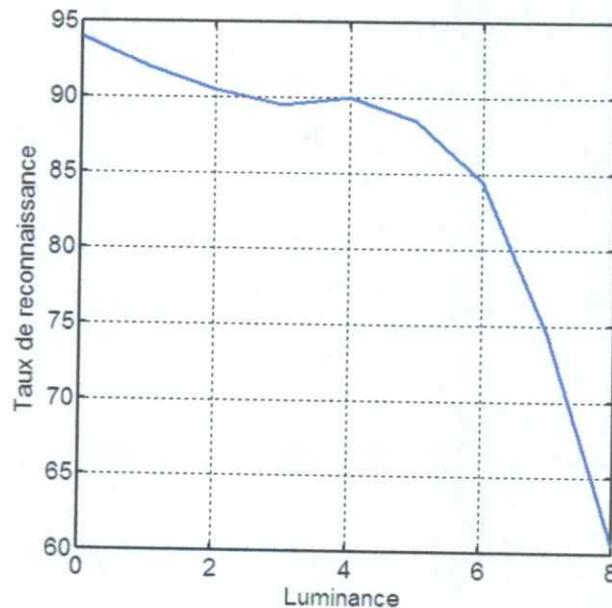


Figure VIII.11 : Taux de reconnaissance en fonction de la luminance

De même que sur la figure VII.6, nous constatons que la luminance altère considérablement le taux de reconnaissance. Nous pouvons affirmer que la (2D)PCA est sensible à la luminance.

VIII.3.4. Dia(2D)PCA :

En appliquant le principe diagonal sur la (2D) PCA, nous obtenons les résultats suivants :

VIII.3.4.1. Vecteurs Dia(2D)PCA et taux de reconnaissance :

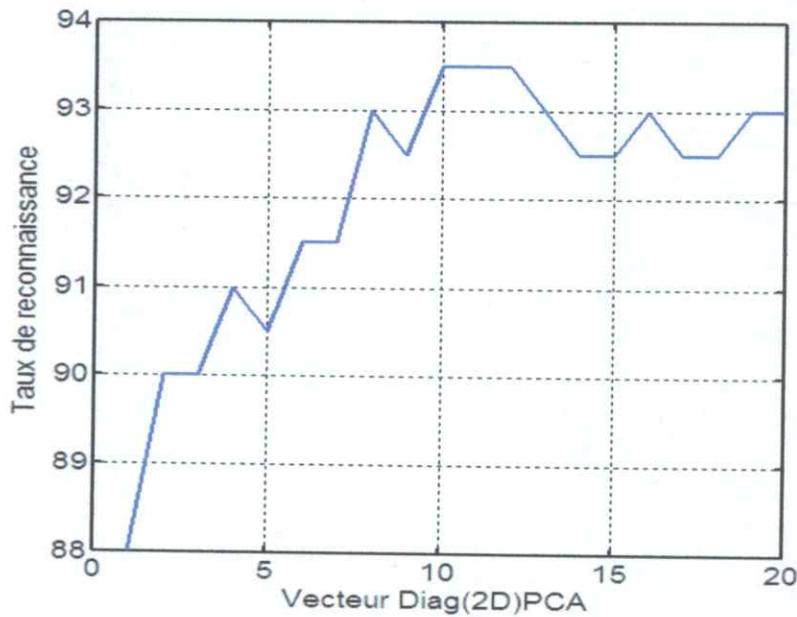


Figure VIII.12: Taux de reconnaissance en fonction du nombre de vecteurs Dia(2D)PCA.

Nous constatons que la forme de la courbe de la Dia(2) PCA est analogue à celle de (2D)PCA avec un taux de reconnaissance maximal de 93.5%, atteint quand le nombre de vecteurs est de 10 ou 11.

VIII.3.4.2. Luminance :

Sur la base de la remarque qui précède, on fixe le nombre de vecteurs à 10 et on varie la luminance de 0 à 80. Nous illustrons l'effet produit sur le taux de reconnaissance dans le graphe qui suit :

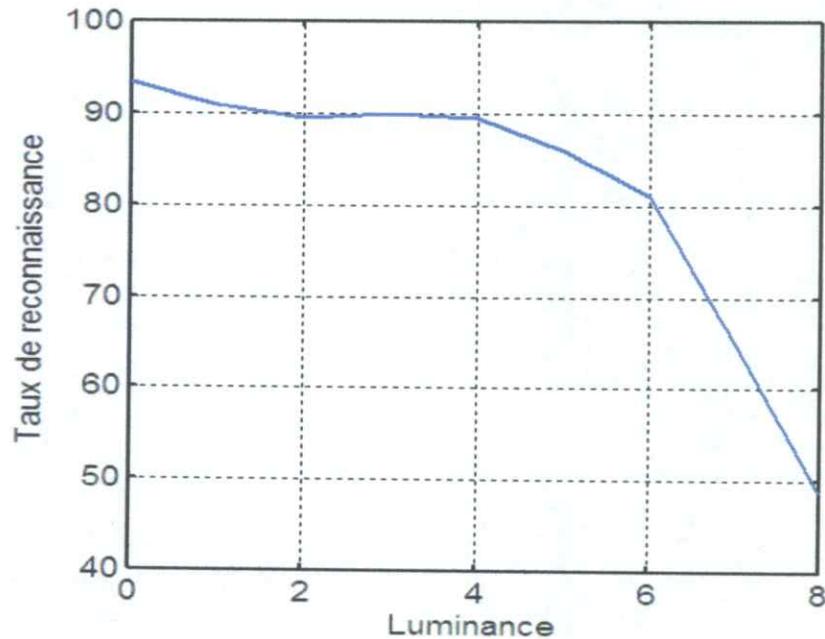


Figure VIII.13: Taux de reconnaissance en fonction de la luminance.

De même que sur la figure VII.6, nous constatons que la luminance altère considérablement le taux de reconnaissance. Nous pouvons affirmer que la Dia(2)PCA est sensible à la luminance.

VIII.3.5. Synthèse :

La comparaison des résultats des variantes de la PCA, nous conduit à affirmer que :

- la diagonale PCA donne de meilleurs résultats que la PCA en raison de son taux de reconnaissance (90.5);
- Les différentes PCA sont sensibles à la luminance, propriété qui nuit à la reconnaissance des visages.
- les approches 2D sont de loin les meilleures, elles requièrent moins de temps d'exécution et atteignent des taux de reconnaissance plus élevés grâce au nombre important d'informations extraites d'une image. En effet, une image de 112x92, projetée avec un vecteur 2DPCA, permet d'obtenir un modèle de taille 112 et un taux de reconnaissance variant entre 87% et 88% tandis qu'avec un vecteur PCA, on obtient un modèle de taille 1 et un taux de reconnaissance qui varie entre 11.5% et 37%. Néanmoins, entre la Dia(2D)PCA et la (2D)PCA, la première est plus stable et la seconde atteint un meilleur taux de reconnaissance, à savoir 94%, ce qui rend cette dernière plus fiable.

Remarque : *Notons que les résultats obtenus varient en fonction de la base d'images sélectionnée.*

VIII.4. Reconnaissance par DCT:

Dans cette partie, nous allons voir si le principe diagonal DCT est meilleur que la DCT dans le domaine de la reconnaissance de visage.

VIII.4.1.1. Effets du chevauchement sur le taux de reconnaissance :

L'effet du chevauchement des blocs 8x8 de la DCT est représenté sur un graphe : nous avons porté le nombre de coefficients de DCT en abscisse et le taux de reconnaissance en ordonnée.

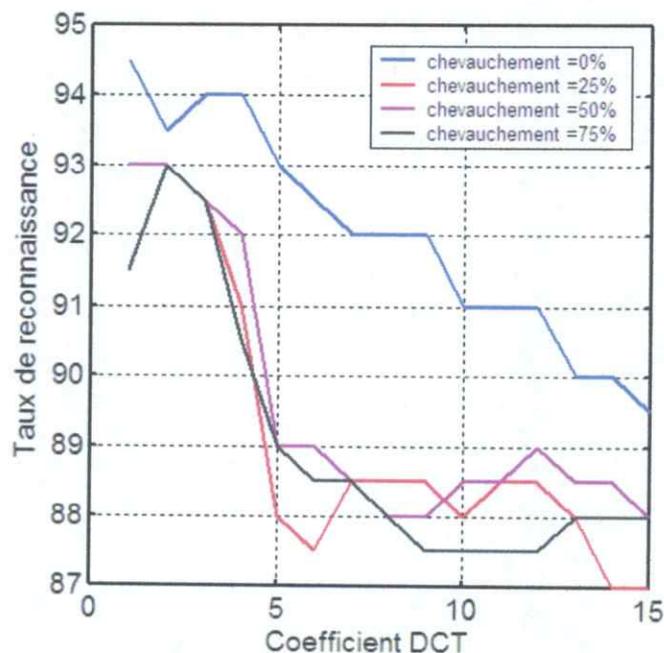


Figure VIII.14: Taux de reconnaissance en fonction du chevauchement et du nombre de coefficients DCT.

Le graphe nous montre qu'un chevauchement de 0% donne de meilleurs résultats. Par conséquent, nous n'aurons plus à recourir à ce paramètre dans ce qui va suivre.

VIII.4.2. Effets de la variation de luminance sur le taux de reconnaissance :

Dans cette partie, nous allons voir, graphiquement, de quelle manière les différentes DCT réagissent à la luminance et ce, afin de les comparer et de dégager la transformation la plus opératoire pour la suite de notre travail.

VIII.4.2.1. DCT Bloc 8x8:

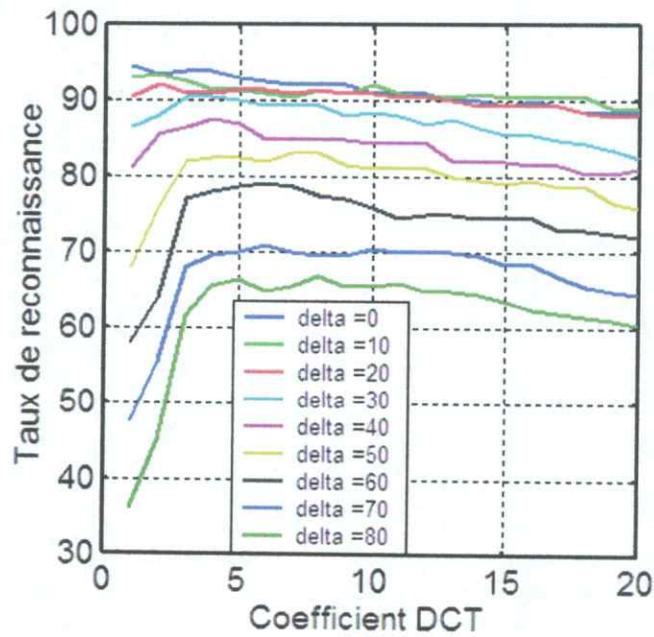


Figure VIII.15: Taux de reconnaissance en fonction de la luminance et du nombre de coefficients DCT.

Le graphe nous montre que de 0 à 30, le taux de reconnaissance n'est pas affecté, mais au-delà de 30, plus la luminance augmente, plus le taux de reconnaissance diminue.

VIII.4.2.2. Diagonal DCT Bloc 8x8:

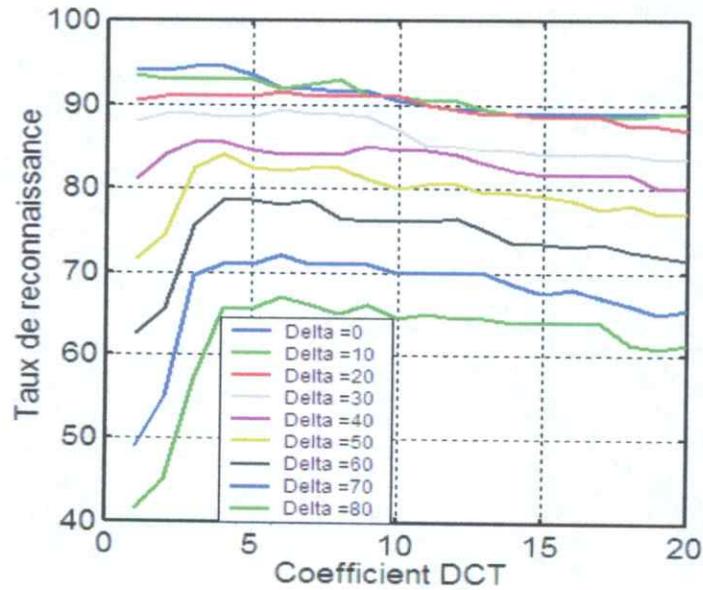


Figure VIII.16: Taux de reconnaissance en fonction de la luminance et du nombre de coefficients DCT.

Même remarque pour la luminance que précédemment et les résultats sont pratiquement les mêmes pour la DCT Bloc et la Diagonale DCT, cependant avec une légère dominance de la DCT Bloc.

VIII.4.2.3. DCT Globale (DCTG) :

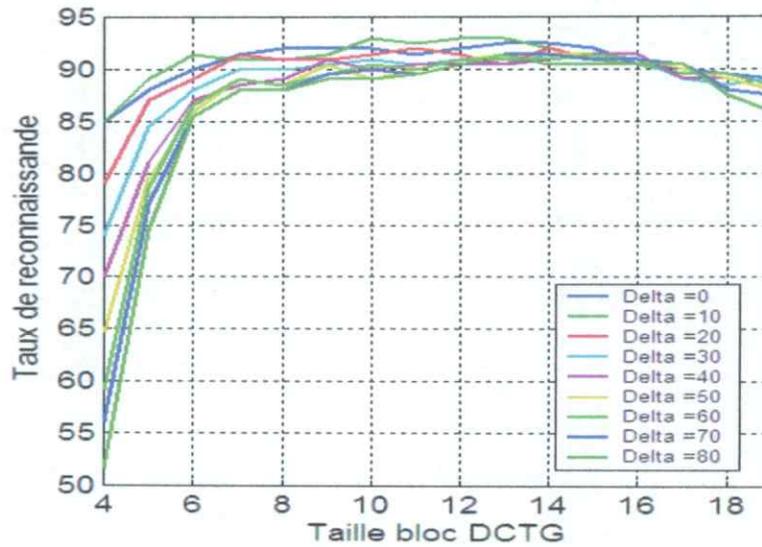


Figure VIII.17: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DCTGlobal.

La luminance affecte faiblement le taux de reconnaissance de la DCTG.

VIII.4.2.4. Diagonale DCTG (DiaDCTG) :

En appliquant la diagonale sur la DCTG, nous obtenons les résultats suivants :

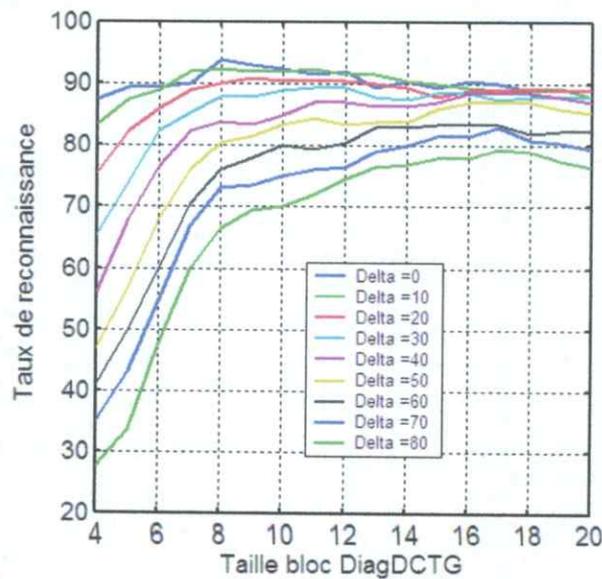


Figure VIII.18: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DiaDCTG.

Le taux de reconnaissance reste élevé (≥ 78) même s'il est affecté négativement par la luminance.

VIII.4.3. Synthèse:

La diagonalisation n'apporte pas d'amélioration conséquente sur la DCT Bloc ni sur la DCTG, mais celle-ci, étant insensible à la luminance, est de loin la meilleure et peut suffire comme classificateur.

VIII.5. Reconnaissance par DCT-PCA :

Nous avons vu les avantages et les inconvénients de la DCT et de la PCA et nous avons essayé tout au long de notre recherche de tirer profit de chacune mais n'arrivant pas à obtenir un bon taux de reconnaissance, nous avons pensé à les combiner, ce que nous avons désigné par la DCT-PCA. Nous allons, donc, voir quel effet va avoir cette combinaison sur la reconnaissance des visages.

VIII.5.1. DCT Bloc8x8-PCA :

VIII.5.1.1. Chevauchement :

Nous ne représenterons ici que l'effet du chevauchement sur le taux de reconnaissance car, comme nous l'avons vu plus haut, la luminance appliquée sur la DCT diminue considérablement le taux de reconnaissance.

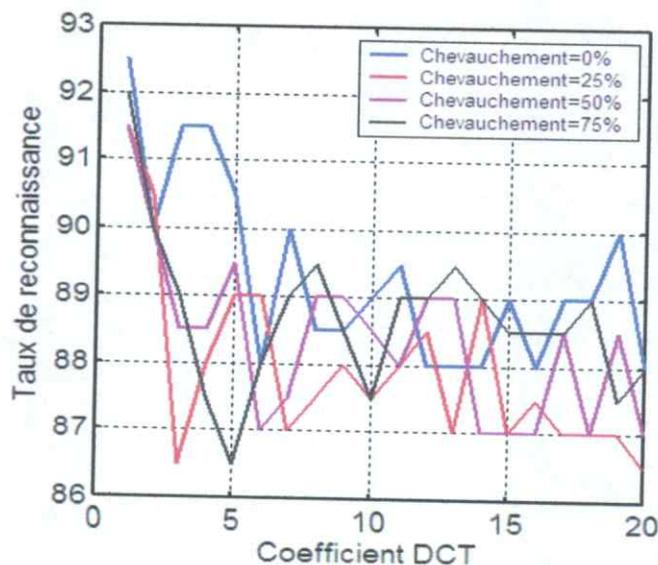


Figure VIII.19: Taux de reconnaissance en fonction du chevauchement et du nombre de coefficients DCT.

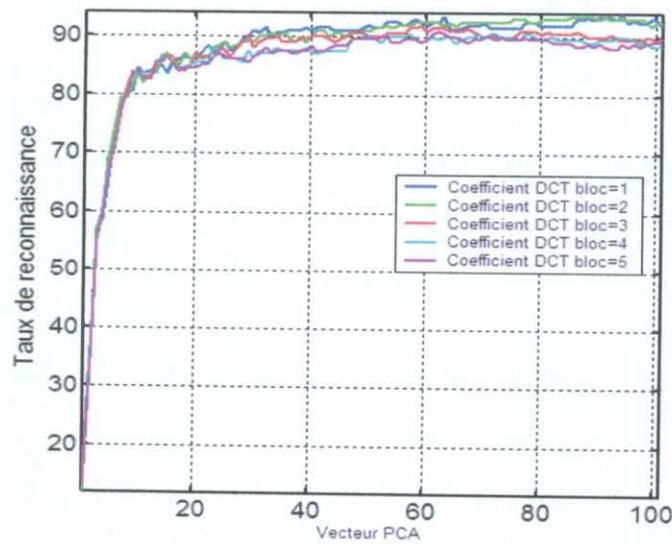


Figure VIII. 20: Taux de reconnaissance en fonction des vecteurs PCA et des coefficients DCTBloc.

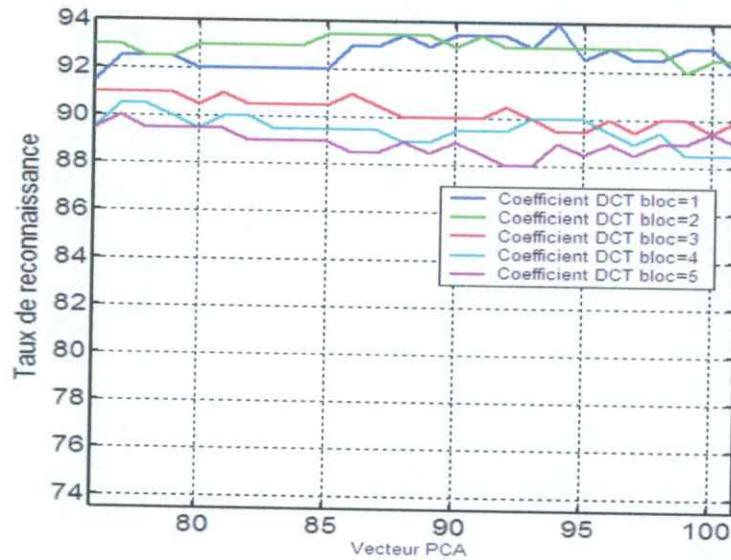


Figure VIII.21: Zoom de la figure VII.20.

Nous constatons que la forme de la courbe de la DCTBloc-PCA est analogue à celle de PCA avec un taux de reconnaissance maximal de 94% atteint quand le nombre de vecteurs est de 94 et un coefficient DCT.

VIII.5.1.2. Luminance :

Sur la base de la remarque qui précède, on fixe le nombre de vecteurs à 94 et on varie la luminance de 0 à 80. Nous illustrons l'effet produit sur le taux de reconnaissance dans le graphe qui suit :

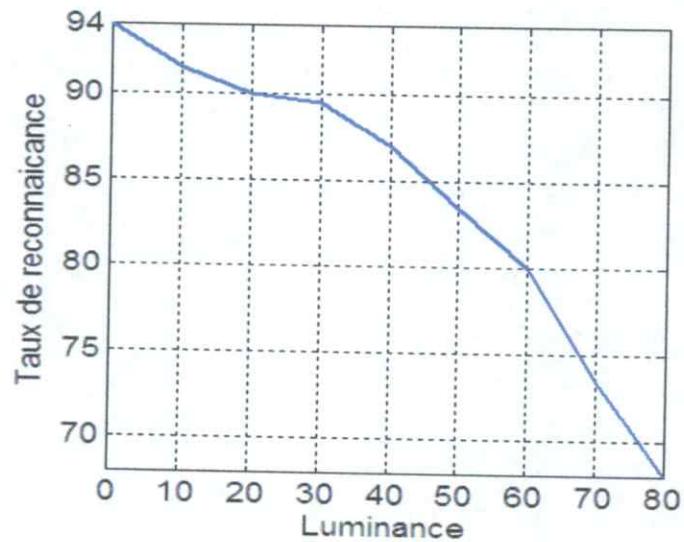


Figure VIII.22: Taux de reconnaissance en fonction de la luminance.

De même que pour les PCA, la DCTBloc-PCA est affectée par la luminance.

VIII.5.2. DCT Bloc8x8 Diag PCA:

VIII.5.2.1. Vecteurs :

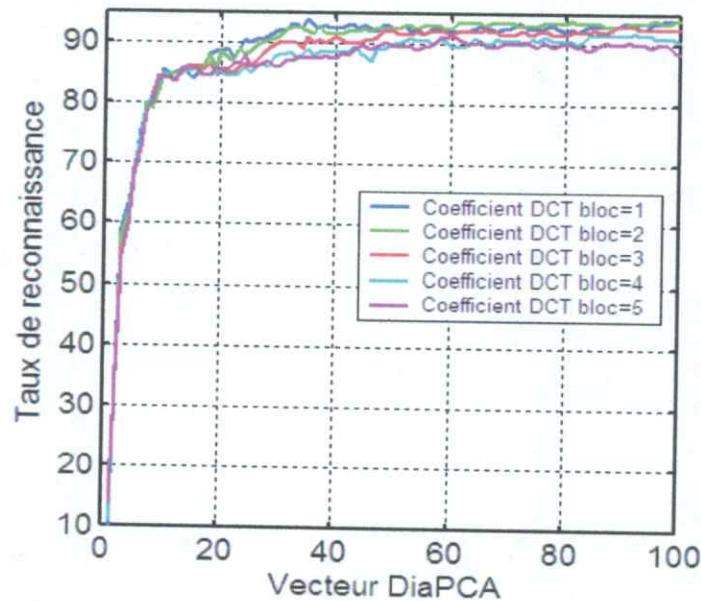


Figure VIII.23: Taux de reconnaissance en fonction des vecteurs DiaPCA et des coefficients DCTBloc.

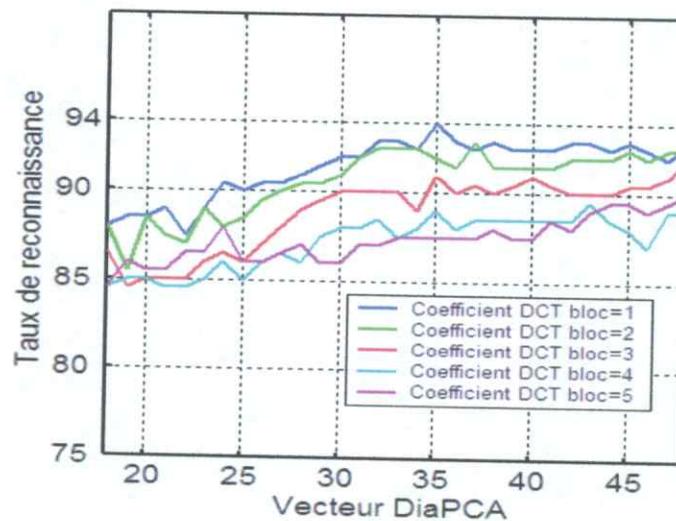


Figure VII.24: Zoom de la figure VII.23.

Nous constatons que la forme de la courbe de la DCTBloc-DiaPCA est analogue à celle de PCA avec un taux de reconnaissance maximal de 94% atteint quand le nombre de vecteurs est de 35 et un coefficient DCT.

VIII.5.2.2. Luminance :

Nous avons fixé le nombre de vecteurs à 35 et avons varié la luminance et nous avons obtenu le graphe suivant :

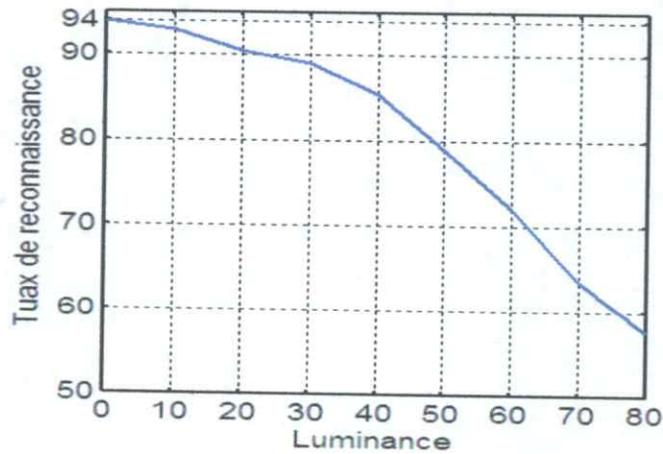


Figure VIII.25: Taux de reconnaissance en fonction de la luminance.

De même que pour les PCA, la DCTBloc-PCA est affectée par la luminance.

VIII.5.3. DCTG (2D)PCA :

Nous avons fixé le nombre de vecteurs à 8 et avons varié la luminance et nous avons obtenu le graphe suivant :

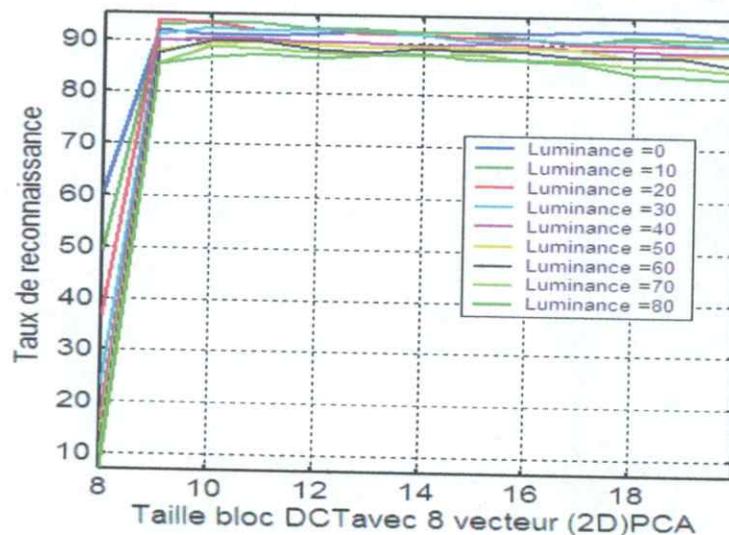


Figure VII.26: Taux de reconnaissance en fonction de la luminance et de la taille du bloc DCT.

Nous constatons que la luminance n'a aucun effet sur le taux de reconnaissance.

VIII.5.3. DCTG-Dia (2D)PCA :

Nous avons fixé le nombre de vecteurs à 8 et avons varié la luminance, nous avons obtenu le graphe suivant :

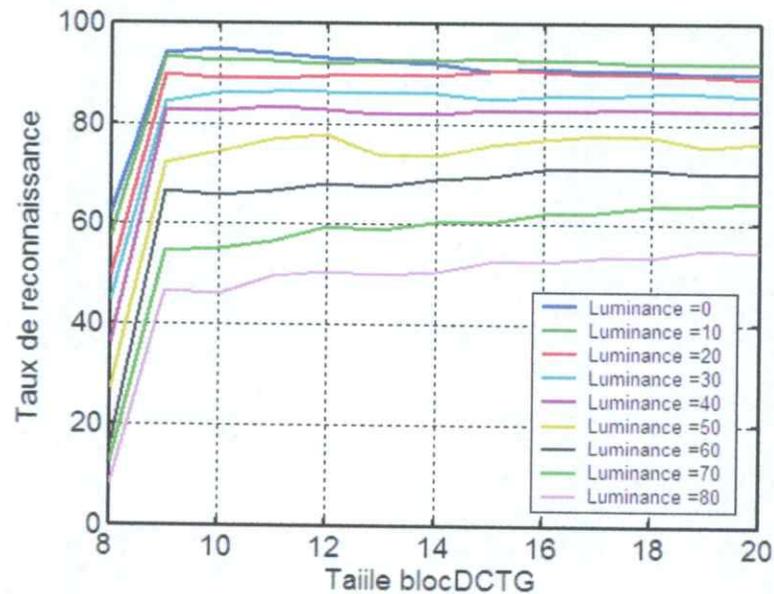


Figure VIII.27: Taux de reconnaissance en fonction de la taille du bloc DCTG et de la luminance.

La DCTG-Dia (2D)PCA atteint le meilleur taux de reconnaissance de 95% mais elle est fortement affectée par la luminance

VII.6. Synthèse :

Nous avons vu que la (2D)PCA est la meilleure des PCA, avec un taux de reconnaissance de 94%, cependant, elle n'admet pas de variation de luminance ; la DCTG est la meilleure des DCT car elle atteint un taux de 94% et elle est insensible à la luminance. La combinaison des deux, autrement dit, la DCTG-(2D) PCA, l'objet de notre recherche, nous donne le même taux de reconnaissance, à savoir, 94%, avec plus de stabilité à la luminance. Par conséquent, nous pouvons affirmer qu'avec cette combinaison nous avons atteint nos objectifs.

CONCLUSION GENERALE

Dans ce mémoire, nous proposons une étude comparative de deux méthodes de reconnaissance de visage. Nous sommes partis de l'hypothèse que si les résultats de la DCT combinée à la PCA étaient satisfaisants, ceux de la DCT combinée à la DiagPCA seraient meilleurs car, théoriquement, la diagonalisation de la PCA donne de meilleurs résultats.

Apports et limites

Nous avons testé séparément les anciennes approches, autrement dit la PCA et la DiagPCA. Mais, ces dernières se sont avérées sensibles à la luminance. Nous avons alors songé à les combiner à la DCT, celle-ci étant moins sensible à la luminance et ayant le pouvoir de réduire la taille des images, ce qui nous donne un gain considérable en temps d'exécution de la PCA.

D'essai en essai, nous sommes arrivés à la DCT Globale-(2D)PCA qui, elle, a donné de meilleurs résultats que toutes les combinaisons que nous avons testées, à savoir un bon taux de reconnaissance (94%) et une meilleure stabilité à la luminance.

Rappelons toutefois que les résultats obtenus sont relatifs à la base de données. En effet, les résultats sont relatifs aux images acquises. Ainsi, si l'image acquise présentait des contraintes (orientation du visage, cheveux, lunettes,...), cela ne faciliterait pas la reconnaissance.

Nous avons, par ailleurs, utilisé d'autres classificateurs (Frobinus, Yang et Assembled Matrix Distance) qui ne figurent pas sur le mémoire, faute d'espace et de temps, mais nous les avons intégrés dans l'application. Ces classificateurs ont donné de très bons résultats avec une base de données plus importante.

Perspectives

Si nous disposions de plus de temps, nous aurions intégré à l'application un module de vérification qui permet de dire si la personne qui se présente est un imposteur ou non, ainsi nous serions passés de l'identification (objet de notre mémoire) à l'authentification (objet de notre ambition).

Actuellement, la reconnaissance de visage n'est fiable que dans des conditions de prise de vue très maîtrisées, ce qui n'écarte pas la possibilité d'une fraude en utilisant, par exemple, une photo d'identité. On pourrait y remédier en recourant à l'imagerie 3D qui pourrait identifier un sujet en mouvement, mal éclairé, se trouvant dans une foule, etc. ou en recourant au « multimodal », autrement dit à une combinaison de plusieurs systèmes biométriques et ce, afin de réduire les risques d'erreurs. Autant de sujets qui pourraient faire séparément l'objet d'une recherche.

ANNEXE :
Rappels mathématiques

Les espaces vectoriels :

Dans cette partie nous allons définir de manière formelle les mécanismes que nous utiliserons plus tard [23-27]. Cette partie est théorique mais les résultats et les propriétés énoncées seront profitables. Nous allons brièvement définir les valeurs et vecteurs propres.

Définitions :**Définition 1 (Espace vectoriel) :**

On appelle espace vectoriel sur K (les scalaires) un ensemble non vide E muni :

D'une loi de composition interne $+$ (c'est la somme de vecteurs), telle que $(E,+)$ soit un groupe commutatif.

D'une loi de composition externe, application de $K \times E$ dans E (c'est le produit par un scalaire, ou encore produit externe) notée $(\lambda, x) \rightarrow \lambda.x$ tel que :

$$\begin{aligned} 1.x &= x \\ (\lambda + \mu).x &= \lambda.x + \mu.x \\ \lambda.(\mu.x) &= (\lambda.\mu).x \\ \lambda.(x + y) &= \lambda.x + \lambda.y \end{aligned} \tag{III-1}$$

Définition 2 (Sous- espace vectoriel) :

Un sous- espace vectoriel de E est une partie non vide F de E :

- Stable pour l'addition de E $((x, y) \in F^2 \Rightarrow x + y \in F)$
- Stable pour le produit externe $((\lambda, x)) \in K \times F \Rightarrow \lambda.x \in F$

(III-2)

Définition 3 (Base d'un espace vectoriel) :

Une famille $(x_i)_{i \in I}$ de vecteurs de E est une base de E si et seulement si elle est libre (ou encore composée de vecteurs linéairement indépendants)

$$\left(\sum_{i \in I} \lambda_i x_i = 0 \right) \Rightarrow (\forall i \in I, \lambda_i = 0) \tag{III-3}$$

et génératrice (tout vecteur de E peut s'écrire comme une combinaison linéaires des vecteurs de la famille)*

$$\forall x \in E, \exists \lambda_i : x = \sum_{i \in I} \lambda_i \cdot x \quad (\text{III-4})$$

Définition 4 (Application linéaire) :

Soient E et F deux espaces vectoriels sur le même corps K . Une application ψ de E dans F est linéaire si et seulement si $\psi(\lambda \cdot x + \mu \cdot y) = \lambda \cdot \psi(x) + \mu \cdot \psi(y)$

Définition 5 :

Une application linéaire est également appelée morphisme d'espaces vectoriels lorsque

- ψ est bijective, c'est un isomorphisme.
- ψ est linéaire de E dans E , c'est un endomorphisme.

Un endomorphisme bijectif est appelé automorphisme.

Enfin, une application linéaire de E dans K s'appelle une forme linéaire sur E .

Définition 6 (Noyau) :

Le noyau d'une application linéaire ψ de E dans F est défini par

$$\ker \psi = \psi^{-1}(0)$$

(C'est l'ensemble des vecteurs de l'espace de départ E qui sont envoyés par ψ dans le 0 de F).

Définition 7 (Image) :

L'image d'une application linéaire ψ de E dans F est définie par

$$\text{im } \psi = \psi(E)$$

(c'est l'ensemble des vecteurs de l'espace d'arrivée F qui sont atteints par ψ).

Comme toute image d'un sous-espace vectoriel par une application linéaire, le noyau d'une application linéaire est un sous-espace vectoriel (de l'espace de départ E).

Comme toute image réciproque d'un sous-espace vectoriel par une application linéaire, l'image d'une application linéaire est un sous-espace vectoriel (de l'espace d'arrivée F).

Définition 8 (Déterminant suivant une base) :

Le déterminant suivant une base $B = (e_1, \dots, e_n)$ de E notée \det_B est la seule forme de E^n dans K qui soit :

- n -linéaire (ses n applications partielles sont linéaires) :

$$\forall (a_1, \dots, a_n) \in E^n, \forall k \in 1, \dots, n : (v \in E) \rightarrow (\det_B(\dots, a_{k-1}, v, a_{k+1}, \dots) \in K) \text{ est linéaire,}$$

- alternée (si on échange deux vecteurs, les signes changent),
- et qui prend la valeur 1 en (e_1, \dots, e_n) .

Notons que toutes les formes n -linéaires alternées ϕ sur E de dimension n sont proportionnelles:

$$\phi(v_1, \dots, v_n) = \phi(e_1, \dots, e_n) \det_B(v_1, \dots, v_n) \quad (\text{III-5})$$

Définition 9 (Déterminant d'un endomorphisme) :

On appelle déterminant d'un ψ de E le scalaire noté \det_ψ défini par endomorphisme

$$\det_\psi = \det_B(\psi(e_1), \dots, \psi(e_n)) \quad (\text{III-6})$$

ou $B = (e_1, \dots, e_n)$ est une base de E .

Notons que ce scalaire est indépendant de la base B .

Définition 10 (Valeur propre, Vecteur propre) :

$\lambda \in K$ est valeur propre de l'endomorphisme $\psi : E \rightarrow E$ s'il existe un vecteur x de E tel que

$$\psi(x) = \lambda \cdot x \quad (\text{III-7})$$

Et x est un vecteur propre associé à λ .

Définition 11 (Spectre) :

On appelle spectre de l'endomorphisme ψ l'ensemble de ses valeurs propres.

Définition 12 (Sous-espace propre) :

Le sous-espace vectoriel défini par $\ker(\psi - \lambda \text{id}_E)$ ou λ est une valeur propre de ψ , est le sous-espace propre associé à λ . Privé de 0, c'est l'ensemble des vecteurs propres associées à λ .

Définition 13 (Polynôme caractéristique) :

On appelle polynôme caractéristique de l'endomorphisme ψ le polynôme de K dans K défini par

$$P_\psi(\lambda) = \det(\psi - \lambda \text{id}_E) \quad (\text{III-8})$$

Où l'on note λid_E l'endomorphisme qui à tout vecteur v associe $\lambda \cdot v$.

Notons que les valeurs propres de ψ sont les racines de son polynôme caractéristique.

Définition 14 (Normes) :

On appelle norme sur E toute application N de E vers R_+ qui a les propriétés suivantes :

- $N(x) = 0 \Leftrightarrow x = 0$
- $\forall \lambda \in K, \forall x \in E : N(\lambda x) = |\lambda| \cdot N(x)$ (III-9)
- $\forall x \in E, \forall y \in E : N(x + y) \leq N(x) + N(y)$ (Inégalité triangulaire)

Un K -espace vectoriel muni d'une norme est appelé espace vectoriel normé.

La norme de x se note en général $N(x)$ ou $\|x\|$

(Toute norme n'est pas une norme euclidienne)

Propriétés :

$$\begin{aligned} \forall x \in E, \|-x\| &= \|x\| \\ \forall x \in E, \forall y \in E : \|x\| - \|y\| &\leq \|x - y\| \leq \|x\| + \|y\| \end{aligned} \quad \text{(III-10)}$$

Exemples :

(Dans E de dimension finie $n > 0$)

Si on se donne une base (e_1, \dots, e_n) , on peut définir pour, $x = \sum_{k=1}^n x_k e_k$ les normes suivantes :

$$\begin{aligned} \|x\|_1 &= \sum_{k=1}^n |x_k| \\ \|x\|_2 &= \sqrt{\sum_{k=1}^n (x_k)^2} \quad (\text{C'est la norme euclidienne}) \\ \|x\|_\infty &= \max_{1 \leq k \leq n} (|x_k|) \end{aligned} \quad \text{(III-11)}$$

Définition 15 (Matrice) :

On appelle matrice de type $(n; p)$ sur un corps K toute application

$$\begin{aligned} M : N_n \times N_p &\rightarrow K \\ (i, j) &\rightarrow m_{i,j} \end{aligned}$$

Définition 16 (Matrice d'une application linéaire dans des bases) :

Soit une application linéaire ψ de E dans F . Soient $B_E = (e_1, \dots, e_p)$ une base de E et $B_F = (f_1, \dots, f_p)$ une base de F . Il existe alors (a_{ij}) tels que $\forall j \in 1, \dots, p$

$$\psi(e_j) = \sum_{i=1}^n a_{ij} f_j \quad \text{(III-11)}$$

La matrice de ψ relativement à B_E et B_F est l'application $Mat \psi_{B_E B_F} : (i, j) \rightarrow (a_{ij})$

Notons que ψ est déterminée de façon unique par $(a_{ij})_{i=1, \dots, n; j=1, \dots, p}$.

Notons également que les colonnes d'une matrice sont les coordonnées dans B_F de l'image par ψ des vecteurs de la base B_E .

Définition 17 (Matrice d'un endomorphisme) :

Les matrices associées aux endomorphismes sont des matrices carrées. La réciproque n'est pas vraie: deux espaces vectoriels différents peuvent avoir la même dimension. Cependant, toute matrice carrée est la matrice d'un endomorphisme (relativement à une base). On peut donc malgré tout définir pour les matrices carrées les notions de déterminant et de spectre. C'est ce que nous allons voir dans les paragraphes suivants.

Considérons une application linéaire endomorphe $\psi : E \rightarrow E$ d'un espace vectoriel E de dimension n sur lui-même. Considérons également deux bases distinctes (bases B_1 et B_2), dans lesquelles F se traduit respectivement par les matrices A et B . Considérons encore la matrice de transformation (ou de passage) T , et rappelons la relation entre les matrices A et B .

$$\begin{aligned} B &= T^{-1} . A T \\ A &= T . B . T^{-1} \end{aligned} \tag{III-12}$$

Définition :

Les matrices carrées A et B sont dites semblables s'il existe une transformation T qui satisfait les relations précédentes.

Nous pouvons nous poser la question de savoir jusqu'où une matrice B se laisse simplifier par le choix d'une base B_2 appropriée. La réponse se trouve dans la théorie des valeurs et vecteurs propres.

Théorème :

La matrice diagonale D , semblable à une matrice quelconque A , est la matrice dont les éléments diagonaux sont les valeurs propres de A , de plus, la matrice de transition (de A vers D) est composée des vecteurs propres de A .

Calcul des valeurs et vecteurs propres :

Comme vu précédemment, la recherche des valeurs et vecteurs propres est une étape indispensable à la diagonalisation d'une matrice. Dans ce domaine, il existe une multitude d'algorithmes qui ont chacun leurs avantages et inconvénients. Cette partie va détailler les

principales méthodes de calcul, les comparer, et en choisir la plus optimale (la mieux adaptée dans notre cas).

Remarque :

Dans cette partie, nous allons nous focaliser sur les matrices symétriques. De plus, la performance de l'algorithme pour des matrices de grande taille sera un élément déterminant quant au choix de la méthode qui sera utilisée. L'intérêt porté sur ces deux paramètres paraîtra plus clair ultérieurement.

Au préalable, nous verrons la tridiagonalisation qui est une transformation de matrices qui s'avérera fort utile ultérieurement.

La tridiagonalisation d'une matrice (la réduction de Householder) :

La tridiagonalisation d'une matrice consiste en une transformation d'une matrice symétrique quelconque en une matrice de la forme suivante :

$$\begin{pmatrix} \alpha_1 & \beta_2 & 0 & \dots & \dots & \dots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & 0 & & & 0 \\ 0 & \beta_3 & \alpha_3 & \beta_4 & 0 & & 0 \\ 0 & & & \dots & & & 0 \\ 0 & & & & \dots & & 0 \\ 0 & & 0 & \beta_{n-1} & \alpha_{n-1} & \beta_n & \\ 0 & & & 0 & \beta_n & \alpha_n & \end{pmatrix} \quad (\text{III-13})$$

Cette transformation se fait grâce à la matrice de Householder.

Définition :

On appelle matrice de Householder toute matrice de la forme :

$$P = I - 2w.w^T \quad (\text{III-14})$$

Où w est un vecteur réel normalisé, $w^T.w = |w|^2 = 1$.

Vérifions que la matrice P est une matrice orthogonale

$$P^2 = (I - 2w.w^T).(I - 2w.w^T)$$

$$= I - 4w.w^T + 4w.(w.w^T).w^T$$

$$= I$$

Cela implique que $P = P^{-1}$. En utilisant la définition de P , on vérifie facilement que $P = P^T$, et on a donc bien construit une transformation orthogonale.

Nous allons maintenant appliquer à P le vecteur x constitué de la première colonne de A . Pour cela, on exprime la matrice P sous la forme suivante :

$$P^T = 1 - \frac{u.u^T}{H}$$

Avec $H = |u|^2 / 2$. Si on choisit le vecteur u tel que :

$$u = x \mp |x|e_1$$

Où e_1 est le vecteur colonne unitaire tel que seule la première composante est non nulle et égale à 1. On obtient facilement la valeur de H

$$H = 2(|x|^2 \pm |x|x_1)$$

En appliquant P à x :

$$P.x = x - \frac{u}{H} \cdot (x \mp |x|e_1)^T . x$$

$$= x - \frac{2u \cdot (|x|^2 \mp |x|x_1)}{2|x|^2 \mp 2|x|x_1}$$

$$= x - u$$

$$= \pm |x|e_1$$

(III-

15)

Construction de la matrice de Householder :

Le résultat précédent montre que la matrice P annule tous les éléments du vecteur x hormis le premier.

La stratégie pour construire les matrices de Householder est la suivante : on choisit un vecteur x constitué des $n-1$ derniers éléments de la première colonne pour construire la matrice P_1 .

En conséquence, on obtient la structure suivante pour P_1 :

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & & & & \\ 0 & & P_1^{(n-1)} & & \\ \dots & & & & \\ 0 & & & & \end{pmatrix} \quad \text{(III-16)}$$

En appliquant la transformation orthogonale à la matrice A , on obtient :

$$A' = P.A.P$$

$$A' = \begin{pmatrix} a_{11} & k & 0 & \dots & 0 \\ k & & & & \\ 0 & & & & \\ \dots & & & & \\ 0 & & & & \end{pmatrix} \quad (\text{III-17})$$

Où k est au signe près la norme du vecteur $(a_{21}, \dots, a_{n1})^T$.

On choisit la seconde matrice de Householder avec un vecteur x qui est constitué avec les $(n-2)$ derniers éléments de la seconde colonne.

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & & & \\ \dots & \dots & P_2^{(n-2)} & & \\ 0 & 0 & & & \end{pmatrix} \quad (\text{III-20})$$

La tridiagonalisation de la partie supérieure de la matrice précédente est préservée, on construit ainsi colonne par colonne une tridiagonalisation de la matrice A . La procédure est complète après $(n-2)$ transformations similaires de Householder.

En pratique, pour éviter la multiplication de matrices, très coûteuse numériquement, on peut exprimer le produit $P.A.P$ en introduisant la notation suivante :

$$p = \frac{Au}{H}$$

En conséquence, la première multiplication matricielle peut être écrite comme suit :

$$A.P = A \cdot \left(1 - \frac{u.u^T}{H}\right)$$

$$= A - p.u^T$$

De même pour la seconde :

$$A' = P.A.P = A - p.u^T - u.p^T + 2Ku.u^T$$

$$\text{Avec } K = \frac{u^T \cdot p}{2H}$$

Enfin, en posant $q = p - Ku$

La matrice A' s'exprime alors simplement :

$$A' = A - q.u^T - u.q^T \quad (\text{II-21})$$

Les différentes méthodes de calcul des valeurs et vecteurs propres :

Les méthodes directes :

Ce premier type de méthodes consiste à considérer les valeurs propres comme étant les racines du polynôme caractéristique de la matrice A ($P_A(\lambda) = \det(A - \lambda I)$)

Si I est la matrice identité. Si A et I sont de taille $n \times n$, les valeurs propres sont donc solutions de l'équation polynomiale de degré n :

$$(-1)^n \cdot (\lambda^n - C_{n-1} \lambda^{n-1} + \dots - C_1 \lambda - C_0) = 0$$

Dans le cas d'une matrice symétrique $n \times n$, l'équation précédente admet n racines réelles, distinctes ou non. Un certain nombre de méthodes anciennes (Leverrier, Krylov) forment le polynôme caractéristique et déterminent ses zéros (cela reste toujours dans le cadre de la résolution de $f(x) = 0$). Ces méthodes sont abandonnées, car, en l'absence d'hypothèses supplémentaires sur A , elles sont lentes (calcul des C_i) et instables (recherche des racines).

Des méthodes itératives ont été établies pour répondre à ces problèmes.

Les méthodes itératives :

Ces méthodes sont principalement appliquées à des matrices de grande taille. Les calculs croissent dans ce cas linéairement avec n . Pour toute méthode itérative, il convient de s'assurer que la convergence est suffisamment rapide pour que le temps de calcul ne soit pas consommé sans que la recherche d'une solution ne soit réellement effectuée.

La méthode de Jacobi :

La méthode de Jacobi revient à effectuer une suite de transformations similaires orthogonales. Chaque transformation est une simple rotation planaire qui permet d'annuler un élément de la matrice A initiale.

La rotation élémentaire P_{pq} est donnée par la matrice :

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & 1 & \dots & \dots & \dots & 0 \\ \dots & \dots & c & 1 & \dots & s & \dots \\ \dots & \dots & \dots & \dots & 1 & \dots & \dots \\ \dots & \dots & -s & \dots & \dots & c & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 1 \end{pmatrix} \quad (\text{III-21})$$

Avec la condition que

$$c^2 + s^2 = 1.$$

Soit la matrice A' telle que

$$A' = P_{pq}^T \cdot A \cdot P_{pq}$$

En notant les coefficients de la matrice a_{ij} , on obtient après calculs que

$$a'_{rp} = ca_{rp} - sa_{rq}$$

$$a'_{rq} = ca_{rq} + sa_{rp}$$

$$a'_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2csa_{pq}$$

$$a'_{qq} = s^2 a_{pp} + c^2 a_{qq} + 2csa_{pq}$$

$$a'_{pq} = (c^2 - s^2)a_{pq} + cs(a_{pp} - a_{qq})$$

Avec $r \neq p$ et $r \neq q$

Si on annule le terme a'_{pq} , en introduisant l'angle de rotation ϕ , ($c = \cos(\phi)$, $s = \sin(\phi)$) on a le rapport θ

$$\theta = \frac{c^2 - s^2}{2sc}$$

$$= \cot(2\phi)$$

$$= \frac{a_{qq} - a_{pp}}{a_{pq}}$$

Si on appelle $t = s/c$, on obtient en utilisant l'équation précédente :

$$t^2 + 2t\theta - 1 = 0$$

La plus petite des racines correspond à un angle de rotation inférieur à $\pi/4$ et donne la méthode la plus stable numériquement. Cette racine peut s'exprimer sous la forme :

$$t = \frac{\text{sgn}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}$$

En utilisant $c^2 + s^2 = 1$, on obtient

$$c = \frac{1}{\sqrt{1+t^2}} \text{ et } s = tc$$

En imposant que le terme a'_{pq} s'annule, on a finalement les relations suivantes :

$$a'_{pp} = a_{pp} - ta_{pq}$$

$$a'_{qq} = a_{qq} + ta_{pq}$$

$$a'_{rp} = a_{rp} - s(a_{rp} + ta_{rq})$$

$$a'_{rq} = a_{rq} + s(a_{rp} + ta_{rq})$$

Avec τ défini par

$$\tau = \frac{s}{1+c}$$

En calculant la somme S suivante

$$S = \sum_{r \neq s} |a_{rs}|^2$$

On peut obtenir une estimation de la convergence de la méthode. Pour une transformation similaire élémentaire, on a :

$$S' = S - 2|a_{pq}|^2$$

Ainsi la suite des transformations conduit à faire décroître la contribution des éléments non diagonaux. Comme la transformation est orthogonale, la somme des carrés des éléments de la matrice est conservée, car la somme des carrés de la diagonale augmente proportionnellement à ce qui est perdu par les éléments non diagonaux. Ainsi formellement, on peut choisir les éléments de la matrice A dans n'importe quel ordre (de préférence on annule toujours le plus grand élément en valeur absolu hors diagonale) et on obtient une méthode qui converge vers une matrice diagonale. Au terme de cette procédure, on a

$$D = V^T . A V \tag{III-22}$$

Où D est une matrice diagonale contenant les différentes valeurs propres et V est une matrice contenant les vecteurs propres correspondants.

La méthode QR :

Cette méthode s'appuie sur le théorème suivant :

Théorème :

Quelque soit la matrice carrée d'ordre n A , il existe deux matrices Q , R telles que :

$$A = Q.R$$

Q est une matrice orthogonale

R est une matrice triangulaire supérieur.

Avant de comprendre comment trouver les valeurs et vecteurs propres grâce à cette méthode, il faut savoir décomposer une matrice en sa forme $Q R$

La décomposition QR :**La décomposition QR avec l'algorithme de Gram Schmidt :**

Cet algorithme se base sur la proposition suivante :

Proposition :

Soit E un espace vectoriel de dimension finie n et $\{v_1, \dots, v_p\}$ ($p \leq n$) une famille de vecteurs linéairement indépendants (on ne peut pas écrire l'un en fonction des autres), on pose alors :

$$\begin{cases} w_1 = \frac{v_1}{\|v_1\|_2} \\ w_q = \frac{\tilde{w}_q}{\|\tilde{w}_q\|_2} \end{cases}$$

$$\text{Avec } \tilde{w}_q = v_q - \sum_{i=1}^{q-1} \langle v_q | w_i \rangle \cdot w_i$$

Alors on a

- 1) $\forall j \in \{1, 2, \dots, p\} w_j \neq \vec{0}$ et $w_j \in L(\{v_1, \dots, v_p\})$
- 2) La famille $\{w_1, \dots, w_p\}$ est orthonormée
- 3) $L(\{w_1, \dots, w_p\}) = L(\{v_1, \dots, v_p\})$

De là, on peut déduire l'algorithme suivant :

L'algorithme :

Soit A une matrice régulière ($\det(A) \neq 0$) alors les vecteurs $\{\vec{a}_1, \dots, \vec{a}_n\}$ (ou $\vec{a}_j = \{a_{1j}, \dots, a_{nj}\}$) forme une base de R^n .

On procède par une orthogonalisation de ce système par Gram Schmidt.

$$1) \text{ On note } r_{11} = \|\vec{a}_1\|_2$$

$$2) \vec{q}_1 = \frac{1}{r_{11}} \vec{a}_1$$

$$3) \text{ Pour } m = 2, 3, \dots, n \text{ et pour } i = 1, 2, \dots, m-1$$

$$r_{im} = \langle \vec{a}_m | \vec{q}_i \rangle$$

$$\tilde{a}_m = a_m - \sum_{i=1}^{m-1} r_{im} q_i$$

$$r_{mm} = \|\tilde{a}_m\|_2$$

$$q_m = \frac{1}{r_{mm}} \tilde{a}_m$$

$$r_{jm} = 0 \quad \forall j = m+1, \dots, n$$

Par suite

$$(\tilde{a}_1, \dots, \tilde{a}_n) = (\tilde{q}_1, \dots, \tilde{q}_n) \cdot (r_{ij})_{1 \leq i, j \leq n}$$

(i.e.: $A = Q.R$)

(III-23)



La décomposition QR par Householder :

Nous avons vu précédemment qu'on pouvait tri diagonaliser n'importe quelle matrice symétrique grâce à un algorithme qui utilise la matrice de Householder. Cependant, ce même type de matrice, utilisé autrement, va nous servir à la factorisation QR d'une matrice quelconque (même celles qui ne sont pas symétriques).

L'algorithme :

On pose $A = A_1$ on note alors $\tilde{a}_1 = (a_{11}, a_{21}, \dots, a_{n1})^T$

Soient :

$$v_- = \tilde{a}_1 - \|\tilde{a}_1\|_2 \cdot \tilde{e}_1 \quad (\tilde{e}_1 \text{ est le premier vecteur de la base canonique})$$

$$H_1 = H_{v_-} = I - 2 \frac{v_- v_-^T}{v_-^T v_-} \quad (I \text{ est la matrice identité})$$

On pose alors A_2 comme suit

$A_2 = H_1 A_1$. Cette matrice sera de la forme suivante :

$$A_2 = \begin{bmatrix} \|\tilde{a}_1\| & \dots & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \end{bmatrix}$$

** : élément quelconque.

Arrivés à l'étape k , nous aurons :

$A_k = H_{k-1} \cdot H_{k-2} \dots H_1 \cdot A_1$ et aura la forme suivante :

$$A_k = \begin{bmatrix} ** & \dots & \dots & \dots & \dots & ** \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & \dots & ** & \dots & \dots & ** \\ 0 & \dots & 0 & ** & \dots & ** \\ 0 & \dots & 0 & ** & \dots & ** \\ 0 & \dots & 0 & ** & \dots & ** \end{bmatrix}$$

On note \vec{a}_k le vecteur \mathbb{R}^{n-k+1} , correspondant à la $k^{\text{ième}}$ colonne de A_k et dont la première composante correspond au $k^{\text{ième}}$ élément de la diagonale de A_k .

On pose alors $v_-^{(k)} = \vec{a}_k - \|\vec{a}_k\| \vec{e}_1^k$ ou \vec{e}_1^k correspond au premier vecteur de la base canonique de \mathbb{R}^{n-k+1} .

On définit H_k :

$$H_k = \begin{bmatrix} I_{n-k} & 0 \\ 0 & H_{v_-^k} \end{bmatrix}$$

Avec I_{n-k} la matrice identité de taille $n-k$ et $H_{v_-^k}$ la matrice associée à $v_-^{(k)}$.

Alors, $A_{k+1} = H_k \cdot A_k = H_k \cdot H_{k-1} \dots H_1 A_1$ est de la forme :

$$A_{k+1} = \begin{bmatrix} ** & \dots & \dots & \dots & \dots & \dots \\ 0 & ** & \dots & \dots & \dots & ** \\ 0 & \dots & ** & \dots & \dots & ** \\ 0 & \dots & \dots & ** & \dots & ** \\ 0 & \dots & \dots & 0 & ** & ** \\ 0 & \dots & \dots & 0 & ** & ** \end{bmatrix}$$

Et ainsi, de proche en proche, on arrive à $A_{n-1} = H_{n-2} H_{n-3} \dots H_1 A$, la matrice A_{n-1} est triangulaire supérieure. Le produit des matrices $H_{n-2} H_{n-3} \dots H_1$ est une matrice symétrique et orthogonale (car elle est le produit de matrices symétriques et orthogonales).

Par suite, on a :

$$A = H_1 \cdot H_2 \dots H_{n-1} \cdot A_{n-1}$$

En prenant : $Q = H_1 \cdot H_2 \dots H_{n-1}$ et $R = A_{n-1}$ on a finalement

$$A = Q \cdot R$$

(III-24)

Quel algorithme choisir pour la factorisation $Q.R$?

Factoriser une matrice A sous forme $Q.R$ par le procédé de Gram Schmidt est une opération instable (i.e. : il y a une accumulation d'erreurs d'arrondi importante qui risque de fausser cette décomposition). De plus, pour utiliser cette méthode, il faut d'abord vérifier que les vecteurs composants A sont bien linéairement indépendants. C'est pour ces raisons que nous opterons pour la factorisation grâce à la méthode de Householder.

Maintenant que nous avons choisit l'algorithme de décomposition $Q.R$ adéquat, voyons comment utiliser cette forme pour diagonaliser la matrice de départ A .

Calcul des valeurs et vecteurs propres par la méthode $Q.R$:

Soit A une matrice carrée quelconque

Etape 1 : on pose $A_1 = A$ et on détermine Q_1 et R_1 tel que $A_1 = Q_1 \cdot R_1$. On définit alors $A_2 = R_1 \cdot Q_1$

Etape 2 : on a A_2 , on détermine Q_2 et R_2 tel que $A_2 = Q_2 \cdot R_2$. On définit alors $A_3 = R_2 \cdot Q_2$
.....etc.

Etape K : on a A_k , on détermine Q_k et R_k tel que $A_k = Q_k \cdot R_k$. On définit alors $A_{k+1} = R_k \cdot Q_k$

.....et ainsi de suite, on obtient alors une suite de matrices $(A_k)_{k \in \mathbb{N}^+}$ dont tous les éléments sont semblables à A

$$A_1 = A$$

$$A_2 = R_1 \cdot Q_1 = Q_1^T \cdot A_1 \cdot Q_1$$

.....etc.

$$A_{k+1} = (Q_1 \dots Q_k)^T \cdot A \cdot (Q_1 \dots Q_k)$$

25)

(III-

Les matrices A_k ressemblent de plus en plus à des matrices triangulaires supérieures tandis que les éléments diagonaux de A_k convergent vers les valeurs propres de A .

Quel algorithme choisir pour le calcul des valeurs et vecteurs propres ?

L'erreur encourue sur les valeurs propres après 4 itérations de l'algorithme de Jacobi n'excède pas 10^{-4} . Les vecteurs propres sont nettement moins précis. C'est toujours le cas pour l'algorithme de Jacobi. Une dernière remarque : l'ordre dans lequel apparaissent les valeurs propres et les vecteurs propres associés est arbitraire, de même que le signe des vecteurs propres. De plus, on peut affirmer qu'à partir de matrices de taille supérieure à 10, cette méthode devient très coûteuse en temps de calcul. C'est pour toutes ces raisons que nous allons exclure cette méthode.

Nous avons montré précédemment que, grâce à la construction de $(n-1)$ matrices de Householder (chacune éliminant une colonne sous diagonale), nous arrivons à factoriser une matrice en une matrice orthogonale Q et une matrice triangulaire supérieure R .

De manière analogue, on peut montrer qu'il existe une décomposition de la forme $A = QL$ où Q est une matrice orthogonale et L une matrice triangulaire inférieure. Les transformations de Householder correspondantes consistent à annuler, colonne par colonne, les éléments de la matrice transformée qui sont situés au dessus de la diagonale (et non plus au dessous).

Pour des raisons de minimisation d'erreurs d'arrondi, il est préférable d'utiliser la décomposition QL au lieu de la décomposition QR .

L'algorithme QL est défini par la suite suivante :

$$\begin{aligned} A_k &= Q_k \cdot L_k \\ A_{k+1} &= L_k \cdot Q_k \end{aligned} \tag{III-26}$$

Deux résultats très importants :

1) Pour une itération de l'algorithme précédent sur une matrice quelconque, le coût de calcul est proportionnel à n^3 , tandis que pour une matrice tri diagonale, ce coût est linéaire n .

2) on peut démontrer que la convergence de l'algorithme précédent dépend de la proximité des valeurs propres (plus les valeurs propres sont éloignées les unes par rapport aux autres, plus l'algorithme converge vite). Nous pouvons donc, en créant un décalage entre les valeurs propres accélérer la convergence, et ceci de la manière suivante :

On remarque que A et $A' = A - sI$ ont les mêmes vecteurs propres (ces matrices commutent) et que leurs valeurs propres diffèrent par une translation. Si u et λ sont éléments propres de A , alors :

$$(A - sI).u = A.u - sI.u = (\lambda - s).u$$

Maintenant, grâce aux résultats énoncés ci-dessus, nous sommes capables de présenter un algorithme de calcul des valeurs et vecteurs propres performant :

On commence par tri diagonaliser la matrice de départ, ensuite, on lui applique l'algorithme *Q.L* avec une translation des valeurs propres.

Comment peut-on encore optimiser le calcul des valeurs et vecteurs propres ?

Il y a un paramètre très important dont nous n'avons pas encore discuté, il s'agit du fait que la matrice qu'on va diagonaliser est la matrice de dispersion d'une matrice vecteur image (notée *IV*). Maintenant, est-ce que nous pouvons optimiser le calcul des valeurs et vecteurs propres à l'aide de cette nouvelle hypothèse ? La réponse est oui, et ceci, en utilisant la proposition suivante :

Remarque :

Nous rappelons que la matrice de dispersion *VC* d'une matrice quelconque *A* (pas forcément diagonale) est :

$$VC = A.A^T \quad \text{(III-27)}$$

Proposition :

Soit *A* ($n \times m$) une matrice avec $n > m$. Alors la matrice $A.A^T$ a uniquement *m* valeurs propres qui ne soient pas $\cong 0$. Pour trouver ces valeurs propres et les vecteurs propres correspondants :

- 1) On calcule les valeurs propres et vecteurs propres de $A^T.A$
- 2) Les valeurs propres calculées à l'étape précédente seront égales aux valeurs propres que l'on recherche.
- 3) Pour trouver maintenant le vecteur propre v_i correspondant à la $i^{\text{ème}}$ valeur propre λ_i , on procède comme suit :

$$v_i = \frac{A.v'_i}{\|A.v'_i\|} \quad \text{Avec } v'_i : \text{ le } i^{\text{ème}} \text{ vecteur propre de } A^T.A$$

Grâce à cette proposition nous pouvons affirmer que, si la matrice IV est de taille $n \times m$ est que $n > m$, il serait plus judicieux de calculer la matrice de dispersion de la matrice transposée de IV et de tirer de là les valeurs et vecteurs propres qui nous intéressent réellement (ceux de $IV \cdot IV^T$).

Transformée en Cosinus Discrète par Bloc $N \times N$ (DCT)

Rappels mathématiques

Le passage par la DCT a été l'idée majeure pour la compression JPEG. En effet ce processus appartient à une classe d'opérations mathématiques, tout comme la Transformée de Fourier. Elle permet un changement de domaine d'étude, tout en gardant exactement la même fonction étudiée. Dans notre cas, on étudie une image.

L'application de la DCT, ou d'une Transformée de Fourier fait passer l'information de l'image du domaine spatial en une représentation identique dans le domaine fréquentiel. Pourquoi ce changement de domaine est-il si intéressant? Justement parce qu'une image classique admet une grande continuité entre les valeurs des pixels. Les hautes fréquences étant réservées à des changements rapides d'intensité du pixel, ceux-ci sont en général minimes dans une image. Ainsi on parvient à représenter l'intégralité de l'information de l'image sur très peu de coefficients, correspondant à des fréquences plutôt basses. La composante continue (valeur moyenne de l'image traitée) ayant une grande importance pour l'oeil.

La DCT s'applique à une matrice carrée. Le résultat fournit est représenté dans une matrice de même dimension. Les basses fréquences se trouvant en haut à gauche de la matrice, et les hautes fréquences en bas à droite.

La transformation matricielle DCT étant Orthogonale, elle s'accompagne d'une méthode d'inversion pour pouvoir revenir dans le domaine spatial. Ainsi après avoir fait des modifications dans le domaine fréquentiel, éliminer des variations de l'image quasiment invisibles par l'oeil humain, on retourne à une représentation sous forme de pixels.

Formule pour calculer la DCT sur une matrice $N \times N$

$$DCT(i, j) = \frac{1}{\sqrt{2}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

$$C(x) = \frac{1}{\sqrt{2}}$$

si x vaut 0, et 1 si $x > 0$.

Formule pour calculer la IDCT sur une matrice NxN

$$pixel(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) DCT(i, j) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right)$$

$$C(x) = \frac{1}{\sqrt{2}}$$

si x vaut 0, et 1 si x>0.

Algorithme de calcul pour la DCT

Une bonne manière d'implémenter la DCT par blocs serait de créer une matrice de la transformée de cosinus, C.

$$C_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2j+1)i\pi}{2N}\right) & \text{si } i > 0 \end{cases}$$

Une fois cette matrice créée, on crée la matrice Ct qui correspond à la transposée de C. Ensuite la transformée par bloc 8x8 se réduit à deux multiplications **matricielles** :

$$DCT = C * Pixels * Ct$$

Donc, le coût de calcul la DCT sur un bloc se réduira à 2N multiplications sur des entiers, et 2N additions sur des entiers : cela représente un gain important par rapport à une simple double boucle sur les indices comme pourrait le laisser suggérer la formule initiale de la DCT. On passe d'un coût de $O(N^2)$ à un coût très intéressant de $O(N)$ pour calculer UN bloc.

Définition (distance) :

Il existe plusieurs méthodes pour calculer la distance entre deux individus parmi ces méthodes nous citons :

On a A et B deux matrice de dimension (m*d) et V1, V2 deux vecteurs de taille m

- La distance L1 : entre deux vecteurs

$$d_{L1}(V1, V2) = \sum_{i=1}^m |V1_i - V2_i|$$

- La distance L2 : entre deux vecteurs

$$d_{L2}(V1, V2) = \sum_{i=1}^m (V1_i - V2_i)^2$$

- La distance de FROBINIUS :

$$d_F(A, B) = \left(\sum_{i=1}^m \sum_{j=1}^d (a_{ij} - b_{ij})^2 \right)^{1/2}$$

- La distance de YANG :

$$d_Y(A, B) = \sum_{j=1}^d \left(\sum_{i=1}^m (a_{ij} - b_{ij})^2 \right)^{1/2}$$

- La distance AMD (Assembled Matrix Distance) :

$$d_{AMD}(A, B) = \left(\sum_{j=1}^d \left(\sum_{i=1}^m (a_{ij} - b_{ij})^2 \right)^{\left(\frac{1}{2}\right)^p} \right)^{1/p} \quad (p > 0)$$

REFERENCES SITOGRAPHIQUES

- [R1]:
http://fr.wikipedia.org/wiki/Biom%C3%A9trie#Diff.C3.A9rentes_significations_du_mot_.22biom.C3.A9trie.22
- [R2]: http://www.gerbeaud.com/creation/img_num/imagenum.htm
- [R3]: Microsoft Encyclopédie Encarta 2007
- [R4]: Encyclopédie Wikipedia www.wikipedia.fr.
- [R5]: Antoine Manzanera Cours de TI / ENSTA 2006/07
- [R6]: R.M.Gray « Vector Quantization » -IEEE ASSP magazine - 1991
- [R7]: M.Kunt « Traitement numérique des images » Vol2, 1993.
- [R8]: Alain Boucher « Traitement d'images » Cours de Traitement d'images 2005, Institut de la francophonie pour l'informatique (IFI)
- [R9]: http://bnazarian.free.fr/Cours/IN_GBM_03.PDF
- [R10]: Article « Détections de contours » Université de Grenoble 1
- [R11]: <http://www.biometrie-online.net>
- [R12]: Turk, M.A., Pentland, A.P., "Eigenfaces for recognition". *J. Cognit. Neurosci.* Vol. 3, p.71–86, 1991.
- [R13]: Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J. "Eigenfaces versus fisherfaces: recognition using class specific linear projection". *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 19, No. 7 :p.711–720,1997.
- [R14]: PHILLIPS, P. J. "Support vector machines applied to face recognition". *Adv. Neural Inform. Process. Syst. 11*, p.803–809, 1998.
- [R15]: Richard Lepage : "Reconnaissance d'algues toxiques par vision artificielle et réseau de neurones", mémoire de recherche, université du Québec à Rimouski, 2004.
- [R16]: Jian Yang, David Zhang:" Two-Dimensional PCA" *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 1, January 2004
- [R17]: Daoqiang Zhang, Zhi-Hua Zhou and Songcan Chen "Diagonal Principal Component Analysis for Face Recognition" National Laboratory for Novel Software Technology Nanjing University, Nanjing 210093, Department of Computer Science and Engineering Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China 2006