

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE  
Département d'Electronique

*Projet de fin d'études en vue de l'obtention du Diplôme  
d'Ingénieur d'Etat en Electronique*

**Implémentation en temps réel sur FPGA  
d'une combinaison séquentielle des  
algorithmes PCA et WPD pour la  
reconnaissance de visages**

Réalisé par : **Mr Mohamed El Amine KHEBRI**  
**Mr Merwan BIREM**

Soutenu le 08 Juillet 2010 devant le jury composé de :

<b>Président :</b>	Mr H.BOUSBIA-SALAH	Maître de conférences, classe A	ENP
<b>Promoteur :</b>	M <sup>me</sup> L.HAMAMI	Professeur	ENP
<b>Examineurs :</b>	Mr B.BOUSSEKSOU	Maître assistant, classe A	ENP

**Juillet 2010**

Ecole Nationale Polytechnique, 10 Av. Hassan Badi, El Harrach, Alger, Algérie

## الملخص:

هذا العمل يُركّز على مُعالجة الصُّور في مجال المقاييس الحيوية و تحديدًا في مجال التعرف على الأشخاص باستعمال الوجه. لتحسين الوقت اللازم لتحديد هُويّة الأشخاص و أيضا مردود هذه العملية، قُمنّا باستعمال الخوارزميات التالية : التحليل باستعمال المركبات الرئيسية و التحويل المُوجي. نُستخدم خوارزمية التحويل الموجي لضغط الصور ما يَسمح باستخدام قاعدة بيانات أكبر و كذلك يقوم بتحسين مردود عملية التعرف على الوجوه، أما خوارزمية التحليل باستعمال المركبات الرئيسية فنُستخدم لاستخراج خصائص الوجوه. وقد تم تطبيق هذا النظام على شريحة إف بي جي إيه بُغية تحقيق الهدف الرئيسي من هذه الدراسة و الذي هو تحسّين الوقت اللازم لعملية التعرف على الوجه و التحصل على نظام يعمل تقريبا في الوقت الحقيقي.

**الكلمات المفتاحية :** معالجة الصور، القياسات الحيوية، التعرف على الوجوه، التحليل باستعمال المركبات الرئيسية، التحويل الموجي، مصفوفة البوابات المنطقية القابلة للبرمجة، في إتش دي إل.

## Résumé:

Ce travail s'intéresse au traitement d'images dans le domaine de la biométrie et plus précisément à la reconnaissance de visages. Afin d'améliorer le temps d'exécution s'approchant du temps réel, ainsi que le taux d'identification, nous avons procédé à une combinaison séquentielle de deux algorithmes WPD et PCA. L'algorithme WPD étant utilisé essentiellement pour la compression des images afin de pouvoir utiliser une base de données plus importante et aussi améliorer les taux de reconnaissances et l'algorithme PCA est utilisé pour l'extraction des paramètres d'identification. Le système d'identification ainsi réalisé a été implémenté sur un matériel dédié à savoir une carte FPGA permettant d'avoir un système temps réel portable.

**Mots-clés:** Traitement d'images, Biométrie, Reconnaissance de visages, Analyse en Composante Principale (ACP), Décomposition en Paquets d'Ondelettes (WPD), réseau de portes programmables (FPGA), VHDL.

## Abstract:

This work focuses on image processing in the field of biometrics and specifically the face recognition domain. To improve the time of the recognition and approach real-time processing, and also improve the rate of identification, we used a sequential combination of two algorithms WPD and PCA. WPD algorithm is used primarily for image compression in order to use a largest database and to improve the recognitions rates; the PCA algorithm is used to extract the parameters identification. The identification system has been implemented on a dedicated hardware ie an FPGA board to have a portable real-time system.

**Keys words :** Image Processing, Biometric, Face recognition, Principal Component Analysis (PCA), Wavelet Packet Decomposition (WPD), Field-programmable gate array (FPGA), VHDL.

# *DEDICACE*

*A mes très chers parents*

*A mes sœurs*

*A ma promotrice : Pr L.HAMAMI*

*A mon binôme : Merwan*

*A tous ceux qui me sont chers*

*A mes amis de Boumerdès et de l'ENP*

*A tous ceux qui ont su croire en moi*

*Je dédie ce modeste travail*

*Mohamed El Amine*

# *DEDICACE*

*A ma mère, ma mère, ma mère et mon père*

*A mes frères*

*A ma sœur*

*A tous les membres de ma famille*

*A ma promotrice : L.HAMAMI*

*A mon binôme : Amine*

*A mes amis de l'ENP spécialement du département  
Électronique et Génie Électrique*

*A mes amis de BARAKI*

*A tous ceux qui ont su croire en moi*

*A tous ceux qui me sont chers*

*A tous ceux que je n'ai pas cités et qui sont présents  
dans mes pensées*

*Je dédie ce modeste travail*

*Merwan*

# REMERCIEMENTS

Ce travail a été effectué dans le laboratoire « *Signal & Communications* » avec l'équipe 2 « *Traitement d'images et reconnaissance de formes* » à l'École Nationale Polytechnique d'Alger.

Tout d'abord, nous tenons à remercier tout particulièrement et très chaleureusement notre Directrice de thèse, Pr L HAMAMI, avec qui nous avons eu tant de plaisir à travailler et qui nous a fait profiter de son expérience. Qu'elle trouve ici notre sincère gratitude.

Nous tenons à remercier le Docteur Hicham BOUSBIA-SALAH, Maître de Conférences à l'ENP, d'avoir accepté d'être le président de notre jury de projet de fin d'études.

Nous remercions également le Docteur Boualem BOUSSEKSOU, Maître de Conférences à l'ENP, d'avoir accepté d'examiner notre travail.

Nous remercions l'ensemble de nos enseignants qui nous ont permis d'acquérir assez de connaissances pour en être là aujourd'hui durant toutes nos années d'études.

Nous adressons nos plus sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à l'aboutissement de ce travail, spécialement à « *Mr MAKHLOUFLIA Seddik* » pour ses précieux conseils et orientations, à « *Mr SADOUN* » pour la carte « *FPGA, Spartan-3* » et à « *Mr BABA-ALI Riadh* » de l'USTHB, qui nous a accueilli au sein du laboratoire de son département, et nous a permis l'utilisation de la carte « *FPGA, Virtex-4* ».

Enfin, nous remercions nos parents. Rien n'aurait été possible sans leur soutien, confiance et générosité.

## Table des matières

<b>Liste des Figures .....</b>	<b>5</b>
<b>Liste des Tableaux.....</b>	<b>8</b>
<b>Liste des Abréviations.....</b>	<b>9</b>
<b>Introduction Générale .....</b>	<b>11</b>
<b>Chapitre 1 : Traitement d'images et Etat de l'art de la biométrie. ....</b>	<b>13</b>
1. Introduction [2] .....	13
2. Notions et Définitions [4] .....	15
2.1. La Moyenne .....	15
2.2. Le Contraste .....	15
2.3. Le Voisinage .....	15
3. Traitement d'image numérique [11] .....	15
3.1. Amélioration – Restauration – Correction .....	16
3.2. Analyse .....	17
3.3. Compression – Codage – Transmission.....	18
4. Application.....	18
5. La biométrie.....	18
5.1. Définition .....	18
5.2. Architecture d'un système biométrique [7].....	19
5.3. Comment évaluer les performances d'un système biométrique ? [7].....	21
5.4. La multimodalité [7].....	23
5.5. Applications de la biométrie .....	25
<b>Chapitre 2 : Etude comparative des différentes techniques de la reconnaissance de personnes et justification du choix de la reconnaissance de visages.....</b>	<b>26</b>
1. Introduction.....	26
2. Panorama des différentes biométries .....	26
2.1. Empreintes digitales.....	26
2.2. L'iris .....	28
2.3. Rétine .....	29
2.4. La Voix .....	30
2.5. Signature .....	31
2.6. Formes de la main.....	31
2.7. Dynamique de frappes.....	32
2.8. Reconnaissance de visages .....	32
3. Comparaison entre les différentes méthodes biométriques :.....	35
4. Justification du choix de la reconnaissance de visage.....	36

<b>Chapitre 3 : Généralités sur la théorie des ondelettes.....</b>	<b>37</b>
1. Introduction [19] .....	37
1.1. Définition d'une ondelette [22] .....	38
2. La transformée en ondelettes continues [19] .....	39
3. La transformée en ondelettes discrètes [19].....	40
4. Quelques familles d'ondelettes .....	41
5. Les filtres passe-bandes et les ondelettes [19].....	42
6. La fonction d'échelle [19] .....	43
7. Codage en sous-bande [19] .....	44
8. La transformée en ondelettes discrète.....	45
9. Applications des ondelettes [20] .....	47
9.1. Principe de fonctionnement de la compression DWT .....	48
10. Conclusion.....	50
<b>Chapitre 4 : Analyse en Composante Principale (PCA).....</b>	<b>51</b>
1. Introduction.....	51
2. Rappels Mathématiques [35] .....	51
2.1. La moyenne.....	51
2.2. L'écart type et la Variance .....	51
2.3. La covariance.....	52
2.4. La Matrice de Covariance.....	52
3. Présentation du principe général de la PCA .....	52
3.1. Représentation du nuage des individus dans l'espace $\mathcal{R}^p$ .....	53
3.2. Le Principe général de la PCA.....	53
4. Présentation de l'algorithme PCA pour la reconnaissance de visages [37].....	55
4.1. Phase d'apprentissage .....	55
4.2. Phase de Classification.....	58
5. Inconvénients de la méthode .....	61
6. Conclusion .....	61
<b>Chapitre 5 : Simulation en Matlab des deux algorithmes et de leur combinaison.....</b>	<b>62</b>
1. Introduction.....	62
2. Présentation de la Base de données (BDD).....	62
2.1. Descriptions de la BDD.....	63
2.2. Variations présentes dans les images d'individus.....	63
3. L'Analyse en Composantes Principales (ACP) sur Matlab .....	63
3.1. Etape n°1 : Concaténation des images .....	64
3.2. Etape n°2 : Calcul des visages propres.....	64
3.3. Etape n°3 : Génération de l'espace propre.....	66
3.4. Etape n°4 : Phase de classification.....	67

4.	Mesure des performances du système RV .....	68
4.1.	Mesure du FRR et du FAR .....	68
4.2.	Mesure du taux d'identification.....	71
5.	Décomposition d'image en paquets d'ondelettes (WPD) .....	72
6.	La combinaison séquentielle de WPD et PCA.....	75
6.1.	Mesure du FRR et du FAR .....	75
6.2.	Mesure du taux d'identification.....	76
7.	Amélioration du taux de reconnaissance du système RV.....	76
7.1.	Augmentation de classes d'individus.....	76
7.2.	Ajout d'un second seuil.....	79
8.	Conclusion : .....	81
<b>Chapitre 6 : Implémentation sur FPGA des deux algorithmes et de leur combinaison. ....</b>		<b>82</b>
1.	Introduction.....	82
2.	Les circuits programmables FPGA [38] .....	82
2.1.	Très brève présentation des FPGA.....	82
2.2.	Rôle de l'ingénieur dans le développement FPGA.....	83
3.	Pourquoi les FPGA ? [39] .....	84
3.1.	Performances.....	84
3.2.	Temps de mise sur le marché .....	84
3.3.	Coût.....	84
3.4.	Fiabilité.....	84
3.5.	Maintenance à long terme.....	85
4.	Description du FPGA [41] .....	85
4.1.	Les CLB (configurable logic bloc).....	86
4.2.	Les IOB (input output bloc) .....	87
4.3.	Les différents types d'interconnexions : .....	88
4.4.	L'oscillateur à quartz .....	90
5.	Le langage de description matérielle VHDL [40] .....	90
5.1.	Historique.....	90
5.2.	Pourquoi utiliser le VHDL .....	91
5.3.	Ensemble synthétisable en VHDL .....	91
5.4.	Exemple d'un code VHDL.....	92
6.	Les différentes étapes de la synthèse automatique [40].....	93
7.	L'implémentation des algorithmes PCA, WPD et leur combinaison.....	93
7.1.	Analyse en Composantes Principales.....	93
7.2.	Décomposition en Paquet d'Ondelettes.....	100
7.3.	PCA et WPD.....	102
8.	Synthétisation des blocs .....	103

8.1. Synthétisation du bloc PCA :.....	104
8.2. Synthétisation du bloc WPD : .....	106
8.3. Synthétisation du bloc PCA_WPD :.....	108
9. Conclusion .....	109
<b>Conclusion Générale.....</b>	<b>110</b>
<b>Bibliographie.....</b>	<b>112</b>
<b>Annexes .....</b>	<b>114</b>
A: Combining Classifiers For Face Recognition .....	114

# Liste des Figures

FIGURE 1-1 : (1) PAVAGE (2) ECHANTILLONNAGE.....	13
FIGURE 1-2 : CONVENTION DE NOTATION.....	13
FIGURE 1-3 : RESOLUTION SPATIAL OU ECHANTILLONNAGE.....	14
FIGURE 1-4 : LA QUANTIFICATION D'UNE IMAGE A DES VALEURS DE DYNAMIQUES DIFFERENTES.....	14
FIGURE 1-5 : LA CONNEXITE D'UN PIXEL.....	15
FIGURE 1-6 : LA CHAINE DE TRAITEMENT D'UNE IMAGE.....	16
FIGURE 1-7 : UTILISATION D'UN FILTRE MEDIAN 3x3.....	16
FIGURE 1-8 : DECOMPOSITIONS SUCCESSIVES DES BLOCS (SPLIT) [5].....	17
FIGURE 1-9 : AGREGATION ITERATIVE DES BLOCS SIMILAIRES AU BLOC 1 (MERGE) [5].....	17
FIGURE 1-10 : DETECTION DE CONTOUR UTILISANT LA METHODE DERIVATIF.....	17
FIGURE 1-11 : ARCHITECTURE D'UN SYSTEME BIOMETRIQUE.....	19
FIGURE 1-12 : ENROLEMENT D'UNE PERSONNE DANS UN SYSTEME BIOMETRIQUE.....	20
FIGURE 1-13 : IDENTIFICATION D'UN INDIVIDU DANS UN SYSTEME BIOMETRIQUE.....	20
FIGURE 1-14 : AUTHENTIFICATION D'UN INDIVIDU DANS UN SYSTEME BIOMETRIQUE.....	21
FIGURE 1-15 : ILLUSTRATION DU FRR ET DU FAR.....	22
FIGURE 1-16 : COURBE ROC.....	22
FIGURE 2-1 : UNE TRACE DIGITALE.....	27
FIGURE 2-2 : LES DIFFERENTS POINTS CARACTERISTIQUES D'UNE EMPREINTE DIGITALE.....	27
FIGURE 2-3 : (A) TERMINAISON, (B) BIFURCATION, (C) PONT, (D) LAC ET (E) ILE [8].....	27
FIGURE 2-4 : DETECTION DES MINUTIES DANS UNE EMPREINTE DIGITALE.....	28
FIGURE 2-5 : L'IRIS.....	28
FIGURE 2-6 : (A) EMPLACEMENT DE LA RETINE DANS L'ŒIL HUMAIN (B) : RETINE HUMAINE [14].....	29
FIGURE 2-7 : SPECTRE D'UN SIGNAL VOCAL.....	30
FIGURE 2-8 : METHODE BIOMETRIQUE, GEOMETRIE DE LA MAIN [17].....	31
FIGURE 2-9 : LE SCHEMA GENERAL D'UN SYSTEME DE RECONNAISSANCE DE VISAGE [18].....	33
FIGURE 2-10 : DISTANCE ENTRE POINTS CARACTERISTIQUES POUR LA METHODE LOCALE [19].....	34
FIGURE 2-11 : PRINCIPE GENERAL D'UN SYSTEME DE RECONNAISSANCE DE VISAGES PAR METHODE GLOBALE.....	35
FIGURE 2-12 : COMPARAISON ENTRE LES DIFFERENTES METHODES BIOMETRIQUES.....	36
FIGURE 3-1 : (A) ONDE (B) ONDELETTE. [30].....	37
FIGURE 3-2 : PAVAGE DU PLAN TEMPS-FREQUENCE. (A) TRANSFORMEE DE FOURIER, (B) TRANSFORMEE EN ONDELETTES. [26].....	38
FIGURE 3-3 : EXEMPLES D'ONDELETTES UNIDIMENSIONNELLES ET LEUR TRANSFORMEE DE FOURIER.....	39
FIGURE 3-4 : LOCALISATION DES ONDELETTES DISCRETES DANS L'ESPACE TEMPS-ECHELLE SUR UNE GRILLE DYADIQUE.....	41
FIGURE 3-5 : FAMILLES D'ONDELETTES (A) HAAR (B) DAUBECHIES (C) COIFLET (D) SYMLET2 (E) MEYER (F) MORLET (G) MEXICAN HAT. [28].....	42
FIGURE 3-6 : LE SPECTRE DU SIGNAL $x(t)$ ET DE SA VERSION DILATEE $x(at)$ .....	43
FIGURE 3-7 : LE SPECTRE DES DIFFERENTES ONDELETTES RESULTANTES PAR MISE A L'ECHELLE (DILATATION) DE L'ONDELETTE MERE DANS LE DOMAINE TEMPOREL.....	43
FIGURE 3-8 : COMMENT UN ENSEMBLE INFINI D'ONDELETTES EST REMPLACE PAR UNE FONCTION D'ECHELLE.....	44
FIGURE 3-9 : FRACTIONNEMENT DU SPECTRE DU SIGNAL AVEC UN BANC DE FILTRES ITERE.....	45
FIGURE 3-10 : LE TRAITEMENT DE SIGNAL EN UTILISANT LES ONDELETTES. [30].....	47
FIGURE 3-11 : IMAGE DECOMPOSEE EN BLOCS AVEC TO. [27].....	48
FIGURE 3-12 : SCHEMA GENERAL DE TRAITEMENT D'UNE IMAGE PAR ONDELETTES. [29].....	49
FIGURE 3-13 : ONDELETTE ET FONCTION D'ECHELLE DE HAAR.....	49
FIGURE 3-14 : TRANSFORMEE EN ONDELETTES NIVEAU APRES NIVEAU.....	50

FIGURE 4-1 : MATRICE X DE TAILLE $N \times P$ .....	53
FIGURE 4-2 : REPRESENTATION D'UN INDIVIDU « I » DANS $R_3$ .....	53
FIGURE 4-3 : PASSAGE D'UN ESPACE $R_p$ A UN SOUS-ESPACE $F_K$ ( $K \ll P$ ).....	54
FIGURE 4-4 : PROJECTION DE L'INDIVIDU « $E_i$ » SUR L'ESPACE $F_K$ .....	54
FIGURE 4-5 : (A) IMAGE D'UN VISAGE EN NIVEAUX DE GRIS, (B) : UNE PARTIE DE LA MATRICE GENEREE.....	56
FIGURE 4-6 : UNE VERSION SIMPLIFIEE DE L'ESPACE PROPRE ILLUSTRANT LES QUATRE RESULTATS DE LA PROJECTION D'UNE IMAGE SUR CE DERNIER. [32]. .....	60
FIGURE 5-1 : QUELQUES VISAGES DE LA BDD UTILISEE.....	62
FIGURE 5-2 : IMAGE DU SCRIPTE MATLAB LECT_BASE.M.....	63
FIGURE 5-3 : IMAGE DU SCRIPT MATLAB DE LA FONCTION QANKA.M.....	64
FIGURE 5-4 : IMAGE MOYENNE DE LA BDD CONTENANT 132 PERSONNES.....	65
FIGURE 5-5 : VARIATION DES VALEURS PROPRES DE NOTRE BDD.....	66
FIGURE 5-6 : LES 12 PREMIERS VISAGES PROPRES DE NOTRE BDD.....	66
FIGURE 5-7 : LA VARIATION DU FAR ET DU FRR EN FONCTION DU SEUIL.....	69
FIGURE 5-8 : LA COURBE ROC POUR $P=4$ .....	69
FIGURE 5-9 : LA VARIATION DU FAR ET DU FRR EN FONCTION DU SEUIL.....	70
FIGURE 5-10 : LA COURBE ROC POUR $P=10$ .....	71
FIGURE 5-11 : TAUX D'IDENTIFICATION EN FONCTION DU NOMBRE DE VISAGES GARDES, EN UTILISANT 132x15 IMAGES DE TESTS. ..	72
FIGURE 5-12 : IMAGE ORIGINALE.....	73
FIGURE 5-13 : IMAGE COMPRESSEE AVEC WPD POUR $N=1$ .....	73
FIGURE 5-14 : LA DECOMPOSITION D'UNE IMAGE DE LA BDD DANS LES SEPT NIVEAUX POSSIBLES.....	74
FIGURE 5-15 : LES IMAGES COMPRESSEES VUES AVEC UN ZOOM.....	74
FIGURE 5-16 : COURBES ROC(S) POUR PLUSIEURS NIVEAUX DE COMPRESSION.....	75
FIGURE 5-17 : LES DIFFERENTES COURBES ROC(S) OBTENUES PAR VARIATION DU Z.....	77
FIGURE 5-18 : LES COURBES ROC(S).....	78
FIGURE 5-19 : RESULTAT DE LA SIMULATION.....	81
FIGURE 6-1 : ARCHITECTURE INTERNE D'UN FPGA.....	85
FIGURE 6-2 : CELLULE LOGIQUE (CLB).....	86
FIGURE 6-3 : INPUT OUTPUT BLOCK (IOB).....	87
FIGURE 6-4 : CONNEXIONS A USAGE GENERAL ET DETAIL D'UNE MATRICE DE COMMUTATION.....	88
FIGURE 6-5 : LES INTERCONNEXIONS DIRECTES.....	89
FIGURE 6-6 : LES LONGUES LIGNES.....	89
FIGURE 6-7 L'OSCILLATEUR A QUARTZ.....	90
FIGURE 6-8 : SOUS-ENSEMBLE SYNTHETISABLE.....	91
FIGURE 6-9 : LES RESULTATS DE NOTRE SIMULATION.....	92
FIGURE 6-10 : LES ETAPES DE LA SYNTHESE AUTOMATIQUE.....	93
FIGURE 6-11 : SCHEMA SYNOPTIQUE DE LA ROM CONTENANT L'IMAGE MOYENNE.....	94
FIGURE 6-12 : SCHEMA SYNOPTIQUE DU BLOC CENTRE.....	94
FIGURE 6-13 : SIMULATION DU BLOC CENTRE.....	95
FIGURE 6-14 : SCHEMA SYNOPTIQUE DE LA ROM CONTENANT LES $P=8$ VISAGES PROPRES.....	95
FIGURE 6-15 : SCHEMA SYNOPTIQUE D'UN BLOC MAC.....	96
FIGURE 6-16 : RESULTAT OBTENU PAR SIMULATION DU BLOC MAC.....	96
FIGURE 6-17 : LE BLOC COORD.....	96
FIGURE 6-18 : LE BLOC PART1.....	97
FIGURE 6-19 : SIMULATION DU BLOC PART1.....	97
FIGURE 6-20 : LE BLOC ACP AVEC $P=8$ VP.....	98
FIGURE 6-21 : RESULTAT DE SIMULATION AVEC UNE IMAGE UTILISEE DANS LA PHASE D'APPRENTISSAGE.....	99
FIGURE 6-22 : RESULTAT DE SIMULATION AVEC L'IMAGE 15 DE LA PERSONNE 26.....	99
FIGURE 6-23 : RESULTAT OBTENU AVEC UNE IMAGE D'UN IMPOSTEUR.....	100
FIGURE 6-24 : APPLICATION DE LA WPD SUR UN SIMPLE EXEMPLE.....	101

FIGURE 6-25 : RESULTAT DE COMPRESSION D'IMAGE AVEC ISE. .... 101

FIGURE 6-26 : LE BLOC WPD2. .... 101

FIGURE 6-27 : RESULTAT DE LA COMPRESSION AVEC WPD2. .... 102

FIGURE 6-28 : LE BLOC PCA\_WPD. .... 102

FIGURE 6-29 : RESULTAT DE LA SIMULATION DU BLOC PCA\_WPD..... 103

## Liste des Tableaux

---

TABLEAU 5-1 : MESURE DU FAR ET DU FRR POUR UNE BDD DE 40 PERSONNES AVEC $P=4$ .....	68
TABLEAU 5-2 : MESURE DU FAR ET FRR POUR UNE BDD DE 40 PERSONNES AVEC $P=10$ .....	70
TABLEAU 5-3 : TAUX D'IDENTIFICATION EN FONCTION DU NOMBRE DE VISAGES PROPRES GARDES.....	71
TABLEAU 5-4 : MESURE DU SEUIL QUI DONNE $FAR \cong FRR$ EN FONCTION DE « $N$ ».....	75
TABLEAU 5-5 : LE TAUX D'IDENTIFICATION EN FONCTION DU NIVEAU DE COMPRESSION .....	76
TABLEAU 5-6 : MESURE DU SEUIL QUI DONNE $FAR \cong FRR$ EN FONCTION DE « $Z$ ».....	76
TABLEAU 5-7 : TAUX D'IDENTIFICATION EN FONCTION DU NOMBRE D'ELEMENTS DANS LA CLASSE « $Z$ ».....	78
TABLEAU 5-8 : TAUX D'IDENTIFICATION EN FONCTION DU NOMBRE « $Z$ » CLASSE D'UNE IMAGE ET DU NIVEAU DE COMPRESSION « $N$ ».....	79
TABLEAU 5-9 : LE PRINCIPE DU SECOND SEUIL.....	80
TABLEAU 5-10 : LES RESULTATS OBTENUS POUR UN SEUIL $S_M=4.9$ .....	80

## Liste des Abréviations

---

<b>ACP</b>	: Analyse en Composante Principale.
<b>ASIC</b>	: Application-specific integrated circuit.
<b>BDD</b>	: Base De Données.
<b>BDTI</b>	: Benchmarking, Technical Consulting, and Software, <a href="http://www.bdti.com">www.bdti.com</a> .
<b>CLB</b>	: Configurable Logic Bloc.
<b>CMOS</b>	: Complementary Metal–Oxide–Semiconductor.
<b>CPU</b>	: Central Processing Unit.
<b>DOD</b>	: Department of Defense.
<b>DSP</b>	: Digital Signal Processing.
<b>EDA</b>	: Electronic design automation.
<b>EER</b>	: Equal Error Rate.
<b>FAR</b>	: False Accpetance Rate.
<b>FPGA</b>	: Field-Programmable Gate Array.
<b>FRR</b>	: False Rejection Rate.
<b>FTC</b>	: Failure to Capture.
<b>FTE</b>	: Failure to Enroll.
<b>GPRS</b>	: General Packet Radio Services.
<b>GPS</b>	: Global Positioning System.
<b>GSM</b>	: Le Global System for Mobile Communications.
<b>HDL</b>	: Hardware description language.
<b>ICA</b>	: Independent Component Analysis.
<b>IEEE</b>	: Institute of Electrical and Electronics Engineers.
<b>IOB</b>	: Input Output Bloc.
<b>ISE</b>	: Integrated Software Environment.
<b>JPEG</b>	: Joint Photographic Experts Group.
<b><math>L^2(\mathbb{R})</math></b>	: Espace de Hilbert.
<b>LCA</b>	: Logic cells arrays.
<b>LDA</b>	: Linear Discriminant Analysis.
<b>MAC</b>	: Media Access Control address.
<b>MATLAB</b>	: MATrix LABoratory.
<b>NIST</b>	: National Institute of Standards and Technologies.
<b>NRE</b>	: Non-recurring engineering.
<b>OEM</b>	: Original Equipment Manufacturer.
<b>PCIExpress</b>	: Peripheral Component Interconnect Express.
<b>RAM</b>	: Random-access memory.
<b>ROC</b>	: Receiver Operating Characteristic.
<b>RV</b>	: Reconnaissance de Visages.
<b>SRAM</b>	: Static random access memory.
<b>TF</b>	: Transformer de Fourier.
<b>TO</b>	: Transforme en Ondelettes.
<b>TOD</b>	: Transforme en Ondelettes Discrète.
<b>TTL</b>	: Transistor-Transistor logic.
<b>VHDL</b>	: VHSIC hardware description language, VHSIC: very-high-speed integrated circuit.
<b>VP</b>	: Visages Propres.
<b>WPD</b>	: Wavelet Packet Decomposition.



# Introduction Générale

---

Depuis la création des premiers calculateurs, plusieurs applications de traitement de signal sont apparues, elles occupent aujourd'hui une place prépondérante dans notre société. De nos jours, le traitement numérique de l'information s'impose dans toutes les disciplines comme un élément incontournable, offrant des services aussi nombreux que variés facilitant grandement notre vie quotidienne.

Le traitement d'images est une science basée sur plusieurs disciplines : les mathématiques, les probabilités, l'informatique et les architectures avancées. Il y a une quarantaine d'années, le domaine de la reconnaissance des formes est apparu avec le besoin d'analyseurs automatiques.

L'analyse automatique est potentiellement beaucoup plus rapide que l'analyse manuelle et, en principe, élimine la subjectivité due à l'opérateur ; de plus elle permet la mesure simultanée d'un grand nombre de paramètres ; enfin, elle est indispensable lorsque l'opération est répétitive.

Face aux différentes menaces que subissent les infrastructures gouvernementales, le besoin de la sécurité ne cesse d'augmenter, d'où l'apparition de la biométrie qui remplace les anciennes méthodes de vérification et identification d'individus, qui ont montré leurs faiblesses. Une des méthodes biométriques privilégiée pour une reconnaissance à distance et sans gêne aux utilisateurs est la reconnaissance des visages, il faut noter que deux modes peuvent exister pour cette dernière :

- **Le premier, local** : L'analyse du visage humain est donnée par la description individuelle de ses parties et de leurs relations, Ce modèle correspond à la manière avec laquelle l'être humain perçoit le visage, c'est-à-dire, à nos notions de traits de visage et de parties comme les yeux, le nez, la bouche, etc.
- **Le second, global** : Utilise la région entière du visage comme entrée à l'algorithme de reconnaissance, plusieurs algorithmes existent, on peut citer : PCA, LDA ...Etc. L'inconvénient majeur de ce mode, est l'importante taille de données à traiter, d'où la nécessité d'un matériel dédié (FPGA par exemple) pour assurer un temps réel pour le traitement.

Les algorithmes du mode global peuvent être soit une :

- **Solution Logicielle** : Les algorithmes seront donc développés sur un matériel déjà existant (Ordinateur personnel par exemple) en utilisant un langage de programmation donné (Matlab Par exemple). Cette solution est caractérisée par son faible coût, mais d'un temps de traitement important,
- **Solution Matériel dédié** : Les algorithmes seront implémentés sur un matériel dédié (FPGA par exemple). Cette solution est caractérisée par son coût élevé, mais avec un temps de traitement de l'ordre du temps réel.

Le domaine dans lequel se situe ce travail est celui de la biométrie. Plus précisément, on présentera l'étude théorique et pratique de la reconnaissance de visages pour une méthode globale basée sur l'algorithme « **Wavelet Packets Decomposition**, où WPD » pour la compression des images, et l'algorithme « **Principal Component Analysis**, où PCA » Pour l'extraction des éléments de décision pour la reconnaissance. Ces deux Algorithmes seront étudiés, programmés et simulés séparément et en combinaison séquentielle. Durant cette étude, on variera plusieurs paramètres pour augmenter les performances du système biométrique comme : le taux d'identification et le temps du traitement.

Pour cela, le travail pratique de ce PFE consiste en deux grandes parties :

- L'utilisation de l'outil de programmation « *Matlab* », pour la vérification et la validation des algorithmes, le choix de ce langage est essentiellement pour la facilité offerte pour la programmation. Comme déjà expliqué, ceci est donc une solution logicielle,
- L'utilisation de l'outil de programmation « *Xilinx ISE* » pour l'implantation du programme écrit en VHDL sur une Carte FPGA, ceci est donc une solution matérielle dédiée.

Notre travail sera réparti en six chapitres :

- **Le chapitre 1 :** Dans ce chapitre on donnera, tout d'abord, quelques notions du traitement d'images comme : l'amélioration et l'analyse de l'image. Ensuite, on présentera un état de l'art sur la biométrie sujet principal de ce PFE.
- **Le chapitre 2 :** On présentera dans ce chapitre une étude comparative des différentes techniques de la reconnaissance de personnes et on justifiera le choix de la reconnaissance de visages.
- **Le chapitre 3 :** A travers ce chapitre on donnera quelques généralités sur la théorie des ondelettes, qui est utilisée dans ce PFE pour la compression des images à l'entrée du système biométrique.
- **Le chapitre 4 :** Dans ce chapitre on présentera une étude théorique de l'algorithme « Analyse en Composante Principale » (où PCA en Anglais), qui est utilisé pour l'extraction des éléments de décision.
- **Le chapitre 5 :** Dans ce chapitre on présentera une solution logicielle de la reconnaissance de visages, la programmation est faite sur « *Matlab* » pour les deux algorithmes séparés et combinés séquentiellement, de plus, plusieurs paramètres comme la classe d'individus, le seuil...etc. sont modifiés pour améliorer les performances du système.
- **Le chapitre 6 :** Dans ce chapitre on présentera une solution matérielle dédiée pour la reconnaissance de visages, la programmation est faite avec l'outil « *Xilinx ISE* » en utilisant VHDL comme langage de programmation, ces programmes seront implantés sur une Carte FPGA...

Nous terminerons ce travail par une conclusion générale qui passera en revue tout ce qui a été abordé dans ce mémoire.

# Chapitre 1 : Traitement d'images et Etat de l'art de la biométrie.

## 1. Introduction [2]

Une image numérique est une fonction à support discret et borné. Le support est multidimensionnel (Deux ou trois dimensions) contenant un ensemble de pixels à valeurs discrètes. Ces dernières peuvent être scalaires (Exemple : images en niveaux de gris) ou bien vectorielles (Exemple : imagerie multi-composantes ou couleur).

La gamme de valeurs possibles varie en fonction du type d'image considérée, une image numérique est associée à un pavage rectangulaire de l'espace. Chaque élément du pavage appelé pixel (Figure 1-1) est désigné par ses coordonnées entières.

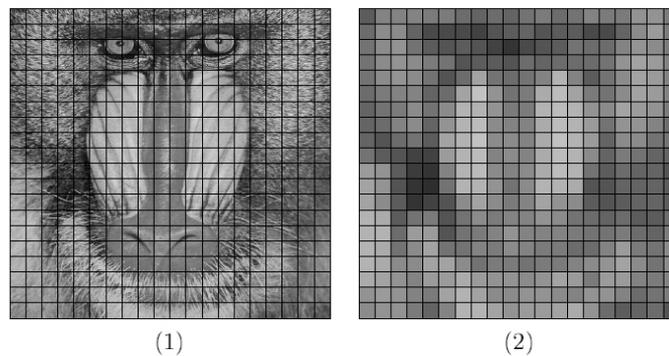


Figure 1-1 : (1) Pavage (2) Echantillonnage.

L'échantillonnage est le procédé de discrétisation spatiale d'une image, il consiste à associer pour chaque pixel une valeur unique, la réduction du nombre de pixels d'une image déjà discrétisée est appelée sous-échantillonnage.

La quantification consiste à limiter le nombre de valeurs différentes que peut prendre chaque pixel de l'image.

Une image numérique est donc une image échantillonnée et quantifiée. Une image numérique de deux dimensions (2D) est représentée par un tableau «  $I$  » de  $H$  lignes et  $W$  colonnes (Voir la Figure 1-2).

Chaque pixel est désigné par un couple de coordonnées  $(i, j)$  où «  $j$  » est l'indice de colonnes et «  $i$  » est celui de lignes.

Par convention le pixel origine  $\{0,0\}$  est en haut à gauche. Le nombre  $I(i, j)$  est la valeur de niveau de gris du pixel  $(i, j)$ , la valeur  $I(i, j) \in \{0, N_{max} - 1\}$  où  $N_{max}$  est le nombre de niveaux de gris possibles.

Le logarithme en base 2 de  $N_{max}$  est appelé « Dynamique de l'image », il représente le nombre de bits utilisés pour coder chaque pixel.

Une image numérique n'est donc qu'une version approchée de l'image réelle, formée par la projection de la scène trois dimensions (3D) sur le plan de la surface photosensible du capteur.

La qualité de l'approximation dépend de la quantité d'informations portées par l'image numérique, mais en particulier du nombre de :

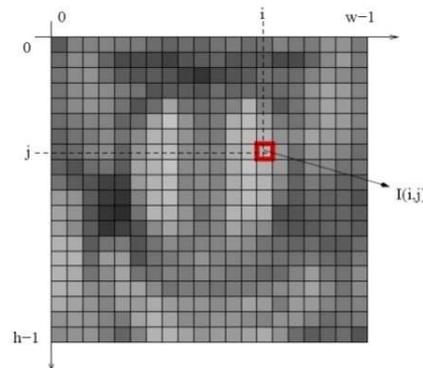


Figure 1-2 : Convention de notation.

- Pixels utilisés définissant la résolution spatiale de l'image (Figure 1-3),
- Bits utilisés représentant chaque pixel. La bonne dynamique dépend de la qualité des éléments photosensibles du capteur, mais aussi de la richesse du contenu informationnel de l'image qui est lié à la distribution de ses valeurs (Figure 1-4).

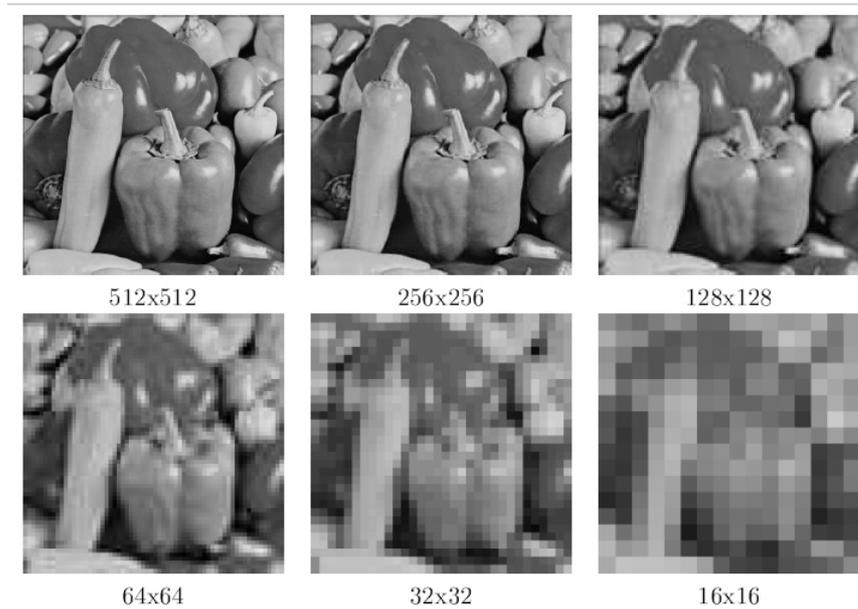


Figure 1-3 : Résolution spatiale ou échantillonnage.

Donc l'image possède deux résolutions :

- La 1<sup>ère</sup> liée au nombre de pixels représentant l'image,
- La 2<sup>ème</sup> liée à la dynamique de l'image, c'est-à-dire le nombre de bits utilisés pour représenter chaque pixel en mémoire.

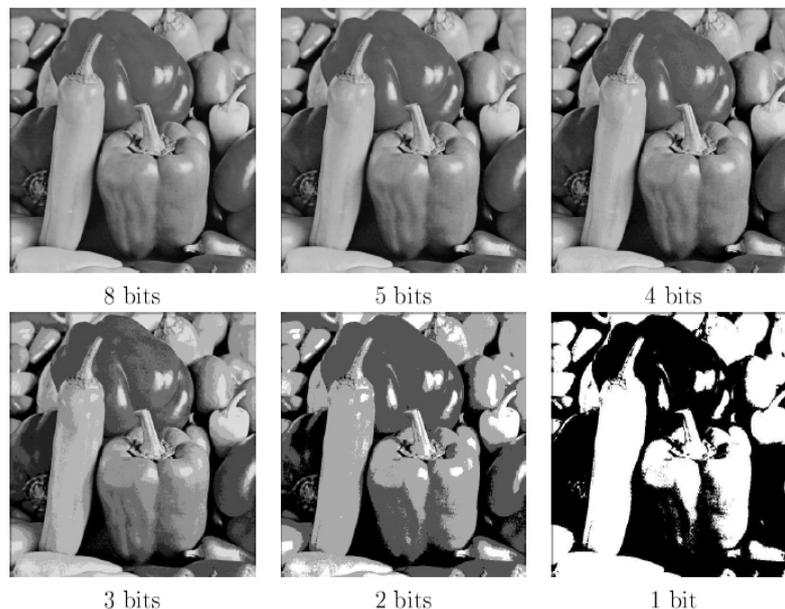


Figure 1-4 : La quantification d'une image à des valeurs de dynamiques différentes.

**Remarque :** Quand l'une des deux résolutions diminue, la taille de l'image en Bits diminue aussi, mais cela engendre une perte d'informations.

## 2. Notions et Définitions [4]

### 2.1. La Moyenne

La moyenne ou luminance (brillance) d'une image est définie comme étant la moyenne de niveaux de gris des pixels de l'image distribués selon une fonction  $f(x, y)$ .

$$Moyenne = \frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \quad (1-1)$$

Où : M et N représentent le nombre de lignes et de colonnes respectivement.

### 2.2. Le Contraste

C'est la différence de niveaux de gris entre deux régions de l'image, plus précisément les régions sombres et claires. Le contraste « C » d'une image peut être défini de plusieurs façons :

- Ecart-type des variations des niveaux de gris de l'image.

$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x, y) - Moy)^2} \quad (1-2)$$

- Variation entre les valeurs minimales et maximales des niveaux de gris de l'image.

$$C = \frac{\max[f(x, y)] - \min[f(x, y)]}{\max[f(x, y)] + \min[f(x, y)]} \quad (1-3)$$

### 2.3. Le Voisinage

Le traitement d'image s'appuie fondamentalement sur des traitements à l'aide de voisinage. Cela signifie que les traitements effectués dans un endroit donné correspondant à un pixel dépendent non seulement de ce pixel mais aussi des pixels appartenant à son voisinage.

Considérons un pixel « P » dont la position dans l'image est donnée par les coordonnées (x, y), un voisinage de « P » noté « V(p) » se définit comme un ensemble de pixels « P' » connectés à « P ». Selon les applications, on peut se placer en 4-connexités ou 8-connexités, cette dernière est utilisée pour la détection de contours et régions car elle offre une meilleure précision (Figure 1-5).

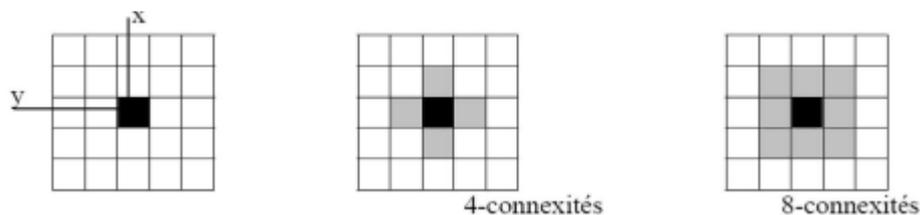


Figure 1-5 : La connexité d'un pixel.

## 3. Traitement d'image numérique [11]

Une image est un signal bidimensionnel portant des informations utiles qu'on veut extraire; pour cela, l'image doit passer par une chaîne de traitement qui utilise certaines particularités de l'image pour faciliter l'analyse et l'interprétation (Figure 1-6).

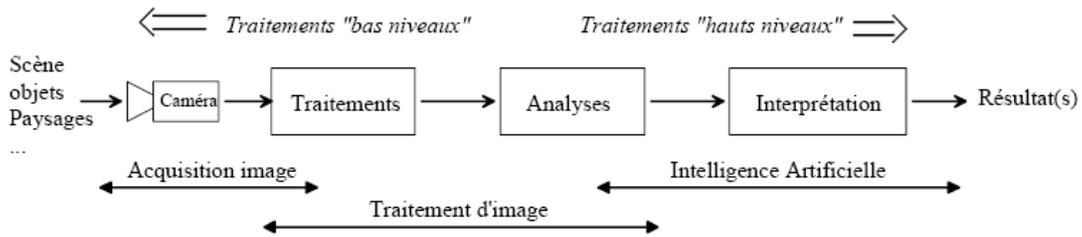


Figure 1-6 : La chaîne de traitement d'une image.

Les traitements qu'une image peut subir sont multiples et peuvent être subdivisés en trois grandes catégories :

### 3.1. Amélioration – Restauration – Correction

Ce traitement améliore le rendu visuel de l'image (Exemple : l'élimination du bruit, l'augmentation ou la diminution de la luminosité de l'image...). Parmi les opérations permettant cela : le filtrage, qui est défini comme étant la convolution de l'image « I » avec un filtre « F », cette opération peut s'écrire sous la forme suivante:

$$(I \otimes F)(x, y) = IF(x, y) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} I(x - n, y - m) \times F(x, y) \quad (1-4)$$

Le filtrage est utilisé essentiellement pour la diminution du bruit provenant du capteur ou de la scène. Ce bruit peut atteindre les zones homogènes ou transitions présentes dans notre image, il existe plusieurs filtres : moyeneur, Gaussien...

Le filtre moyeneur (le plus simple) est le filtre ci-dessous de dimension 3x3, il effectue une moyenne des niveaux de gris des pixels entourant notre pixel central.

$$\frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Il existe un autre type de filtre qui ne nécessite pas de faire une convolution comme le filtre médian, qui opère comme suit :

- Il tri d'abord les pixels de voisinage d'un pixel central,
- Ensuite il affecte la valeur du milieu au pixel central.

Ci-dessous un exemple de l'utilisation du filtre médian dans l'élimination du bruit impulsionnel :



Figure 1-7 : Utilisation d'un filtre médian 3x3.

### 3.2. Analyse

L'analyse permet au système d'être autonome, il pourra donc conclure automatiquement des résultats ou exécuter des commandes lors de la réception des images, par exemple : Reconnaissance d'objets (Dans le cas de notre PFE : Reconnaissance de visages), l'indexation dans une base de données...

Pour assurer la reconnaissance de formes, il faut d'abord que notre système extrait de l'image quelques caractéristiques comme les régions ou les contours en utilisant la segmentation.

Généralement les images sont constituées de plusieurs zones homogènes qui correspondent aux différents objets de la scène. Afin de distinguer ces objets il nous suffit de repérer les différentes régions homogènes en faisant une segmentation de notre image, deux approches existent :

- **Approche région** : Cherche à regrouper directement des pixels ayant des propriétés communes, l'ensemble de ces pixels représente un segment de notre image, parmi les algorithmes utilisés : Le Split & Merge<sup>1</sup> et la croissance de région.

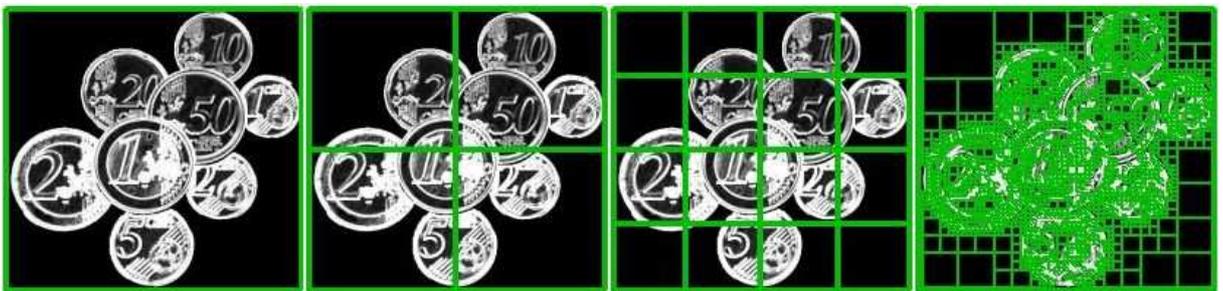


Figure 1-8 : Décompositions successives des blocs (Split) [5].

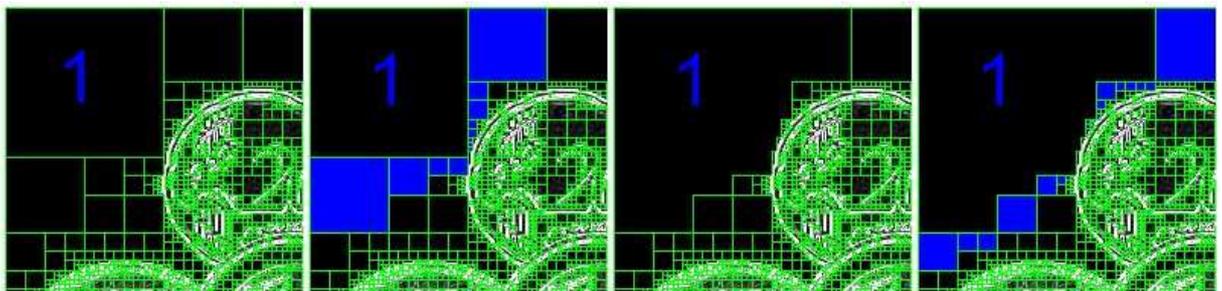


Figure 1-9 : Agrégation itérative des blocs similaires au bloc 1 (Merge) [5].

- **Approche contour** : Détecte les contours des différents objets pour pouvoir délimiter les régions, les algorithmes dérivatifs de premier et second ordre sont souvent utilisés.

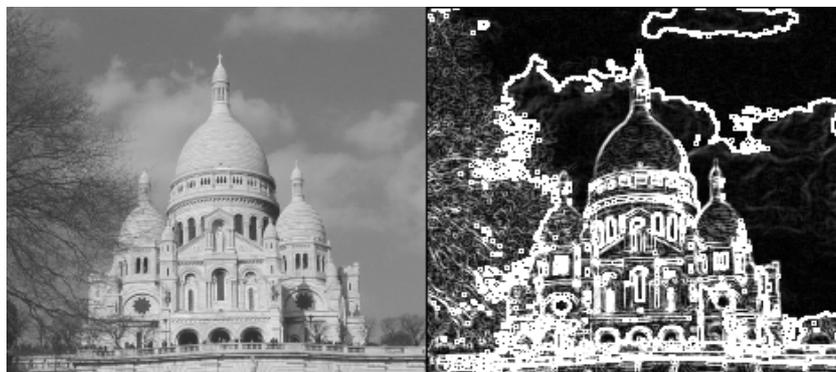


Figure 1-10 : Détection de contour utilisant la méthode dérivatif.

<sup>1</sup> Split & Merge: Décomposition/Fusion.

### 3.3. Compression – Codage – Transmission

---

La transmission d'images numérique nécessite un codage sans perte d'informations qui réduit la taille de l'image pour minimiser le temps de transmission et le volume occupé dans la mémoire.

Le traitement d'images fait appel à plusieurs sciences : l'électronique, l'optique, l'informatique..., et outils : machines de calculs générales (Ordinateur) ou dédiées (FPGA, DSP...).

Dans le cadre de ce PFE, Une carte FPGA sera utilisée pour l'implémentation en temps réel des deux algorithmes PCA et WPD.

## 4. Application

---

Le traitement d'images numériques est utilisé dans plusieurs domaines et dans différentes applications comme :

### **IMAGERIE AERIENNE et SPATIALE**

- Ressources naturelles et humaines.
- Surveillance.
- Météorologie.

### **SCIENCE**

- Interventions en milieux confinés.
- Astronomie.
- Microscope électronique.
- Robotique mobile.
- Biologie.

### **MEDECINE**

- Cytologie.
- Tomographie RMN.
- Echographie.
- Inspection in situ.

### **INDUSTRIE**

- Contrôle non destructif.
- Inspection automatique.
- Mesures automatiques.
- Vision robotique.

### **COMMUNICATION**

- Multimédia.
- Télévision et Vidéo.
- Photographie.
- Edition.
- Indexation

### **DOMAINE MILITAIRE**

- Surveillance.
- Guidage automatique.
- Poursuite d'engins.
- Topographie.

## 5. La biométrie

---

La biométrie est l'une des applications de traitement d'images la plus utilisée, à cause de ses applications qui améliorent la vie de tous les jours en la rendant plus confortable et sécurisée.

### 5.1. Définition

---

La biométrie désigne dans un sens très large l'étude des êtres vivants. Actuellement l'usage de ce terme se rapporte de plus en plus à la reconnaissance des individus à partir de leurs caractéristiques physiques (l'iris, le visage, l'empreinte digitale...) ou comportementales (La voix, l'écriture, le rythme de frappe...).

La biométrie est un nouveau moyen de reconnaissance de personnes qui remplace deux anciennes méthodes qui sont basées sur la :

- *Connaissance* : Mot de passe...etc.
- *Possession* : Un badge...etc.

Ces deux méthodes ont leurs faiblesses, pour la première, le mot de passe peut être oublié par son utilisateur ou deviné par une autre personne. Pour la deuxième, le badge peut être perdu ou volé. L'utilisation de la biométrie pour la reconnaissance de personnes permet d'éviter ce genre de faiblesses car les caractéristiques physiques ou comportementales ne peuvent être oubliées, volées ou perdues. Pour qu'un système soit biométrique utilisé pour la reconnaissance de personnes, ses caractéristiques doivent être basées sur les éléments suivants:

- Universelles : toute la population à identifier doit les posséder,

- Uniques : deux personnes ne peuvent pas posséder les mêmes caractéristiques,
- Permanentes : invariables au cours du temps,
- Performantes : l'identification doit être rapide et précise,
- Acceptées : par les utilisateurs,
- Impossible à dupliquer : par un imposteur,
- Mesurables facilement et en un temps réduit.

## 5.2. Architecture d'un système biométrique [7]

Dans un système biométrique il existe au minimum deux modules, l'un pour la phase d'apprentissage et l'autre pour la phase de reconnaissance, un troisième module peut être ajouté pour assurer une adaptation et une amélioration de notre système (Figure 1-11).

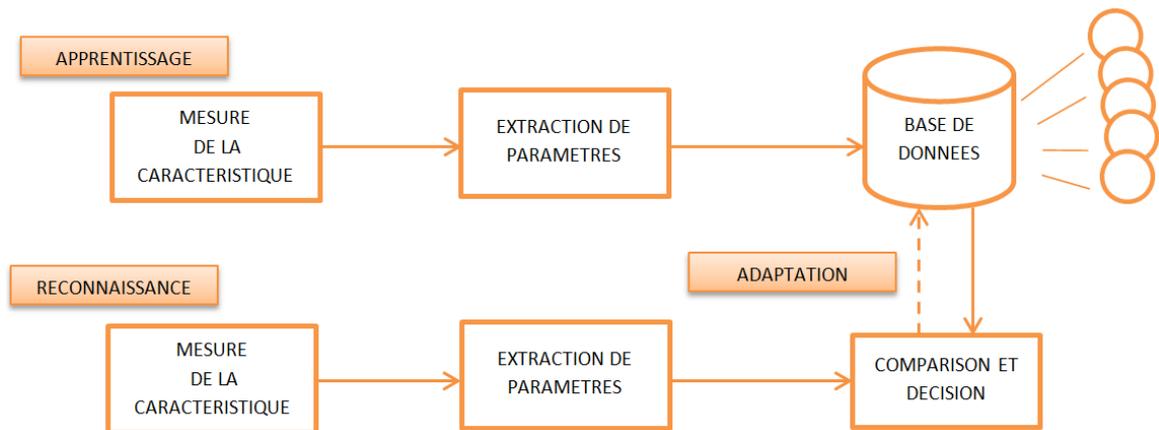


Figure 1-11 : Architecture d'un système biométrique

### 5.2.1. Module d'apprentissage

La principale tâche de ce module est la génération de la base de données qui contiendra un ensemble de paramètres caractérisant les individus à reconnaître par la suite. Ce module effectue les opérations suivantes :

- **L'acquisition** : Assurée par un capteur de préférence de bonne qualité pour réduire le temps d'exécution de l'étape suivante (Exemple : une caméra pour les visages...),
- **Traitement et extraction des zones de décision** : le signal reçu est en général bruité et pas entièrement utile, cette étape éliminera les bruits et les parties non utiles afin de minimiser sa taille et son temps de traitement (Extraction du visage, lèvres, nez, yeux...),
- **Extraction des paramètres caractéristiques** : Après l'extraction des zones de décisions, un modèle plus simple doit être déterminé pour représenter la mesure, il est obtenu par le calcul des paramètres caractéristiques qui varient suivant la méthode de reconnaissance adoptée et le paramètre biométrique utilisé,
- **Stockage** : Le modèle résultant sera stocké dans une base de données et sera utilisé pour effectuer la phase de reconnaissance.

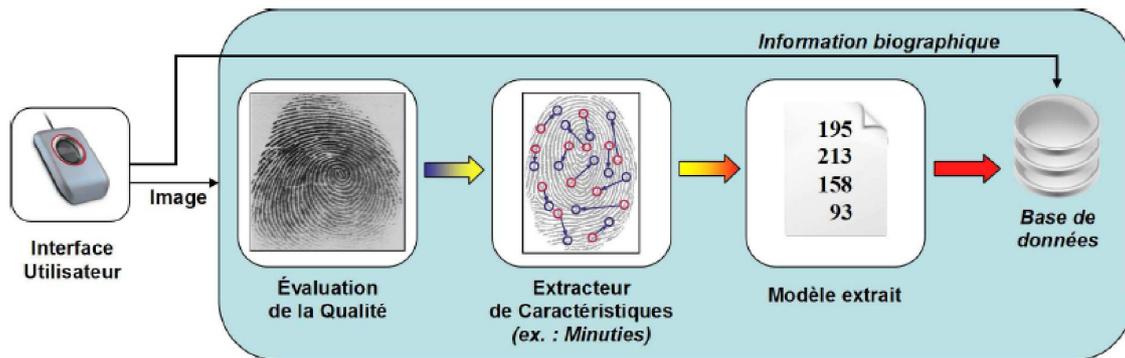


Figure 1-12 : Enrôlement d'une personne dans un système biométrique.

**Remarque :** Comme ce module n'assure que la génération de la base de données, la notion du temps d'exécution n'est pas très importante.

### 5.2.2. Module de Reconnaissance (Classificateur)

Quand un individu se présente sur le système biométrique, ce module décidera si cette personne est acceptée ou bien rejetée selon des critères mis par l'administrateur.

La premier étape est la mesure du caractère biométrique avec un capteur de préférence similaire à celui utilisé dans le module d'apprentissage dans le cas où cela n'est pas possible une chaine de prétraitement est donc nécessaire. Une fois la mesure faite, les mêmes étapes utilisées pour extraire le modèle d'un individu dans le module d'apprentissage seront utilisées pour extraire le modèle représentant le nouvel individu.

Après l'extraction du nouveau modèle, l'étape suivante dépend du type de système de reconnaissance utilisé :

- **Identification :** C'est la comparaison du nouveau modèle avec tous les modèles présents dans la base de données. On peut remplacer le nom de ce mode par la question « Qui suis-je ? » ou « Suis-je présent dans la base de données ? » (Figure 1-13),

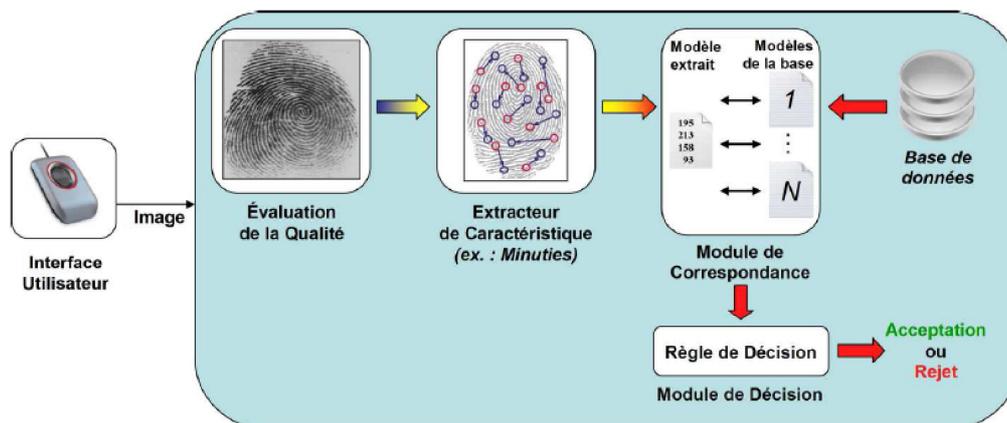


Figure 1-13 : Identification d'un individu dans un système biométrique.

- **Vérification (Authentification)** : C'est la comparaison du nouveau modèle avec un modèle proposé par l'utilisateur. On peut remplacer le nom de ce mode par la question « Suis-je bien la personne que je prétends être ? » (Figure 1-14).

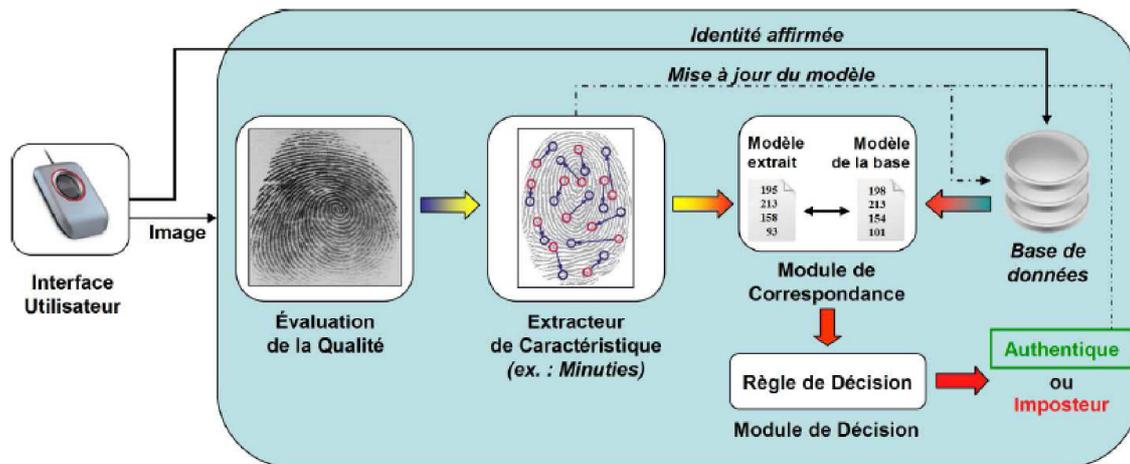


Figure 1-14 : Authentification d'un individu dans un système biométrique.

L'identification et la vérification sont deux méthodes différentes de la reconnaissance. L'identification peut-être une tâche redoutable lorsque la base de données contient trop de modèles, tout particulièrement lorsqu'il existe des contraintes de type « temps réel » sur le système.

### 5.2.3 .Module d'adaptation

Pendant la phase d'apprentissage, le système ne prend que quelques instances d'un même attribut à fin de limiter la gêne pour cet utilisateur. L'accomplissement de la base de données peut se faire par le module d'adaptation qui améliore et maintient les performances du système. Cette amélioration se fait par deux méthodes :

- **Supervisée** : Les données ajoutées dans la base de données sont approuvées par un administrateur,
- **Non-supervisée** : Les données sont ajoutées automatiquement dans la base de données. C'est la méthode la plus utilisée en pratique, mais qui peut engendrer des problèmes en cas d'erreurs du module de reconnaissance.

### 5.3. Comment évaluer les performances d'un système biométrique ? [7]

La performance d'un système de reconnaissance peut se mesurer principalement à l'aide de trois critères :

- Sa Précision,
- Son Efficacité (vitesse d'exécution),
- Son Volume de données nécessaires.

Dans cette section, on ne s'intéressera qu'au critère de « précision ». Tout d'abord, afin de comprendre comment déterminer la précision d'un système biométrique, il nous faut définir clairement trois critères principaux :

1. **Le Taux de Faux Rejet ("False Reject Rate" ou FRR)** : Ce taux représente le pourcentage de personnes censées être reconnues mais qui sont rejetées par le système,

2. **Le Taux de Fausse Acceptation ("False Accept Rate" ou FAR) :** Ce taux représente le pourcentage de personnes censées ne pas être reconnues mais qui sont tout de même acceptées par le système,
3. **Le Taux d'égale Erreur ("Equal Error Rate" ou EER) :** Ce taux est calculé à partir des deux premiers critères et constitue un point de mesure de performance courant. Ce point correspond à l'endroit où FRR est égale au FAR, c'est-à-dire le meilleur compromis entre les faux rejets et les fausses acceptations.

La Figure 1-15 illustre le FRR et le FAR à partir de distributions des scores authentiques et imposteurs tandis que l'EER est représenté sur la Figure 1-16.

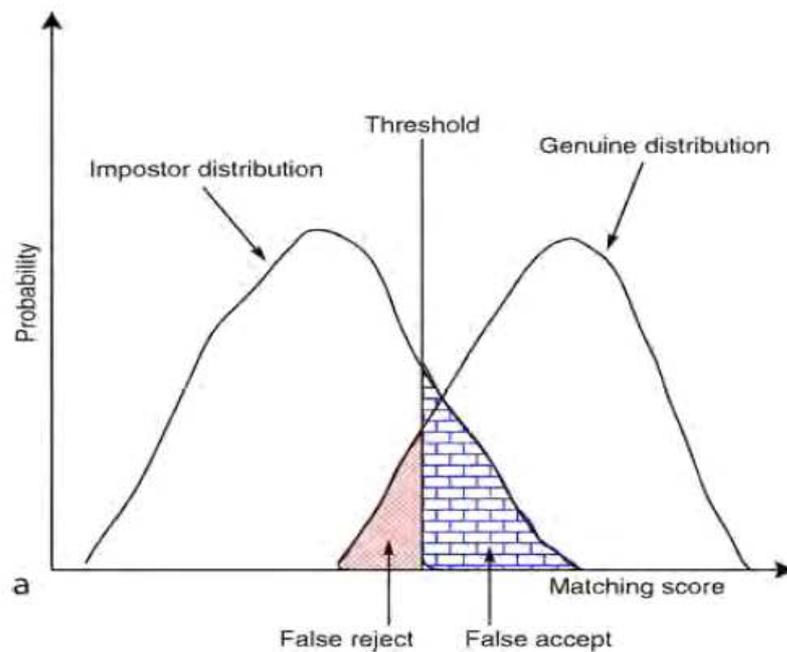


Figure 1-15 : Illustration du FRR et du FAR.

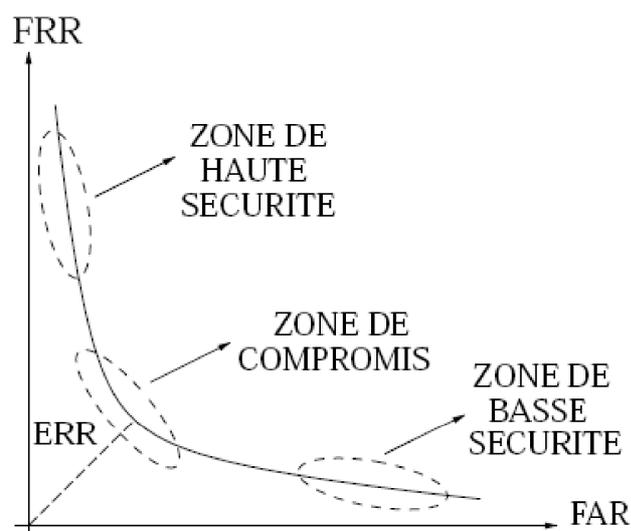


Figure 1-16 : Courbe ROC.

Selon la nature (authentification ou identification) du système biométrique, il existe deux façons pour mesurer la performance :

- En mode authentification, on utilise ce que l'on appelle une courbe ROC ("Receiver Operating Characteristic" en anglais). La courbe ROC trace le taux de faux rejet en fonction du taux de fausses acceptations. Plus cette courbe tend à épouser la forme du repère, plus le système est performant, c'est-à-dire possédant un taux de reconnaissance global élevé,
- En mode identification, après l'extraction des caractéristiques du nouveau visage ces caractéristiques sont comparées avec tout le contenu de la base afin de trouver l'élément le plus proche. Le visage le plus proche au visage de test est celui ayant un score de similarité le plus élevé, le taux d'identification indique le nombre de fois pour lesquelles le visage le plus proche déjà trouvé correspond bien à la personne identifiée. On peut poser la question suivante pour bien comprendre ce taux d'identification : « Est-ce que le visage le plus proche trouvé par le calcul de similarité correspond à la personne qui veut s'identifier ? » ou « Combien de fois une bonne identification c'est produite sur « k » tests ? ».

### 5.4. La multimodalité [7]

---

Bien que les techniques de reconnaissance biométrique promettent d'être très performantes, on ne peut garantir actuellement un excellent taux de reconnaissance avec des systèmes biométriques uni-modaux, basés sur une unique signature biométrique. De plus, ces systèmes sont souvent affectés par les problèmes suivants :

- **Bruit introduit par le capteur** : du bruit peut être présent dans les données biométriques acquises, ceci étant principalement dû à un capteur défaillant ou mal entretenu. Par exemple, l'accumulation de poussière sur un capteur d'empreintes digitales, un mauvais focus de caméra entraînant du flou dans des images de visage ou d'iris, etc. Le taux de reconnaissance d'un système biométrique est très sensible à la qualité de l'échantillon biométrique et des données bruitées peuvent sérieusement compromettre la précision du système.
- **Non-universalité** : si chaque individu d'une population ciblée est capable de présenter une modalité biométrique pour un système donné, alors cette modalité est dite universelle. Ce principe d'universalité constitue une des conditions nécessaires de base pour un module de reconnaissance biométrique. Cependant, toutes les modalités biométriques ne sont pas vraiment universelles. Le National Institute of Standards and Technologies (NIST) a rapporté qu'il n'était pas possible d'obtenir une bonne qualité d'empreinte digitale pour environ 2% de la population (personnes avec des handicaps liés à la main, individus effectuant de nombreux travaux manuels répétés, etc.). Ainsi, de telles personnes ne peuvent pas être enrôlées dans un système de vérification par empreinte digitale. De la même manière, des personnes ayant de très longs cils et celles souffrant d'anomalies des yeux ou de maladies oculaires ne peuvent fournir des images d'iris, ou de rétine, de bonne qualité pour une reconnaissance automatique. La non-universalité entraîne des erreurs d'enrôlement ("Failure to Enroll" ou FTE) et/ou des erreurs de capture ("Failure to Capture" ou FTC) dans un système biométrique.
- **Manque d'individualité** : les caractéristiques extraites à partir de données biométriques d'individus différents peuvent être relativement similaires. Par exemple, une certaine partie de la population peut avoir une apparence faciale pratiquement identique due à des facteurs génétiques (père et fils, vrais jumeaux, etc.). Ce manque

d'unicité augmente le taux de fausse acceptation ("False Accept Rate" ou FAR) d'un système biométrique.

- **Manque de représentation invariante** : les données biométriques acquises à partir d'un utilisateur lors de la phase de reconnaissance ne sont pas identiques aux données qui ont été utilisées pour générer le modèle de ce même utilisateur lors de la phase d'enrôlement. Ceci est connu sous le nom de "variations intra-classe". Ces variations peuvent être dues à une mauvaise interaction de l'utilisateur avec le capteur (par exemple, changements de pose et d'expression faciale lorsque l'utilisateur se tient devant une caméra), à l'utilisation de capteurs différents lors de l'enrôlement et de la vérification, à des changements de conditions de l'environnement ambiant (par exemple, changements en éclairage pour un système de reconnaissance faciale) ou encore à des changements inhérents à la modalité biométrique (par exemple, apparition de rides dues à la vieillesse, présence de cheveux dans l'image de visage, présence de cicatrices dans une empreinte digitale, etc.). Idéalement, les caractéristiques extraites à partir des données biométriques doivent être relativement invariantes à ces changements. Cependant, dans la plupart des systèmes biométriques, ces caractéristiques ne sont pas invariantes et par conséquent, des algorithmes complexes sont requis pour prendre en compte ces variations. De grandes variations intra-classe augmentent généralement le taux de faux rejet ("False Reject Rate" ou FRR) d'un système biométrique,
- **Sensibilité aux attaques** : bien qu'il semble très difficile de voler les modalités biométriques d'une personne, il est toujours possible de contourner un système biométrique en utilisant des modalités biométriques usurpées. Des études ont montré qu'il était possible de fabriquer de fausses empreintes digitales en gomme et de les utiliser pour contrer un système biométrique. Les modalités biométriques comportementales telles que la signature et la voix sont plus sensibles à ce genre d'attaque que les modalités biométriques physiologiques.

Ainsi, à cause de tous ces problèmes pratiques, les taux d'erreur associés à des systèmes biométriques uni-modaux sont relativement élevés, ce qui les rend inacceptables pour un déploiement d'applications critiques de sécurité. Pour pallier ces inconvénients, une solution est l'utilisation de plusieurs modalités biométriques au sein d'un même système, on parle alors de « *système biométrique multimodal* ».

Parmi les formes de multimodalité existantes dans le domaine de la reconnaissance biométrique, on cite :

1. **Système multiple biométrique** : Combinaison de deux méthodes biométriques différentes, par exemple : on utilise la méthode de la reconnaissance du visage et celle de la voix,
2. **Système multiple d'acquisition** : L'utilisation de deux capteurs différents, par exemple : Scanners (l'un optique et l'autre thermique) pour les empreintes digitales,
3. **Augmentation du nombre d'unités biométriques mesurées** : par exemple faire la mesure de deux Iris au lieu d'un seul,
4. **Instance multiple d'une même mesure** : Faire une capture répétée pour le même attribut biométrique et avec le même système d'acquisition,

5. **Algorithme multiples** : Utiliser différents algorithmes de reconnaissance sur le même signal d'entrée.

### 5.5. Applications de la biométrie

---

On peut distinguer quatre grands types d'applications de la biométrie : le contrôle d'accès, l'authentification des transactions, la répression et la personnalisation.

- **Contrôle d'accès** : Le contrôle d'accès est lui-même subdivisé en deux sous-catégories :
  - *Contrôle d'accès physique* : Utilisé quand l'utilisateur tente d'accéder à un lieu sécurisé (comme dans les aéroports),
  - *Contrôle d'accès virtuel* : Utilisé quand l'utilisateur tente d'accéder à une ressource ou un service, par exemple pour accéder à un réseau d'ordinateurs ou à l'administration d'un site web. De nos jours, cette application biométrique n'est pas très utilisée, ce domaine étant dominé par les systèmes d'identification basé sur une connaissance, mais elle devra avoir une popularité croissante s'il y aura une chute des prix des systèmes d'acquisitions.
- **Authentification des transactions** : Englobe les retraits d'argent au guichet des banques, les paiements par carte bancaires, les transferts de fond, les paiements effectués à distance par téléphone ou sur internet,
- **Répression** : Une des applications de la biométrie à la répression est la criminologie en utilisant la reconnaissance des empreintes digitales. Il existe aussi des applications dans le domaine judiciaire,
- **Personnalisation** : La biométrie apporte un confort d'utilisation en personnalisant les appareils qu'on utilise tous les jours, par exemple la voiture, ordinateur...etc.

# Chapitre 2 : Etude comparative des différentes techniques de la reconnaissance de personnes et justification du choix de la reconnaissance de visages.

---

## 1. Introduction

---

L'utilisation de la biométrie pour la reconnaissance de personnes est de plus en plus nécessaire pour accroître le niveau de sécurité. D'ailleurs en 2009 les autorités Algériennes ont opté pour un passeport et une carte d'identité biométriques. Mais il faut noter qu'aucune méthode biométrique n'est idéale.

A travers ce chapitre, nous présenterons quelques techniques de reconnaissance de personnes pour les caractéristiques :

- **Physiques (Statiques)** : La reconnaissance de visages, d'empreintes digitales, de la géométrie de la main, de la rétine et de l'iris.
- **Comportementales (Dynamiques)** : La reconnaissance de la voix, la signature et la dynamique de frappes.

Il existe d'autres méthodes qui ne sont pas présentées dans ce chapitre, elles sont basées généralement sur les veines de la main, L'A.D.N, l'odeur corporelle, la forme des oreilles ou des lèvres, la démarche .... [8]

On conclura le chapitre avec une comparaison entre les différentes méthodes biométriques et on présentera nos justifications pour le choix de la reconnaissance de visages pour ce PFE.

## 2. Panorama des différentes biométries

---

### 2.1. Empreintes digitales

---

Une empreinte digitale est le dessin formé par les lignes de la peau des doigts, des paumes des mains, des orteils ou de la plante des pieds. [12]

On peut définir une trace digitale (Figure 2-1) par le résultat d'un contact entre un doigt et un support, elle peut être :

- **Visible** : Dite positive lors de l'apposition de matière et elle est dite négative lors de l'enlèvement de matière,
- **Latente (invisible à l'œil nu)** : La trace vient du dépôt de sueur et/ou du dépôt de sécrétions sébacées,
- **Moulée** : Lorsque la trace vient du contact d'un doigt avec une surface malléable (cire, mastic...). [13]



Figure 2-1 : Une trace digitale.

Les empreintes digitales sont composées de points caractéristiques utiles pour la biométrie (Figure 2-2 et Figure 2-3), on peut citer les :

- **Terminaisons en crêtes** : Soit les points où les crêtes s'arrêtent.
- **Bifurcations** : Soit les points où les crêtes se divisent en deux.
- **Le noyau** : Soit le point intérieur, situé en général au milieu de l'empreinte. Il sert souvent de point de repère pour situer les autres minuties.

Une empreinte complète contient en moyenne une centaine de points caractéristiques mais les contrôles ne sont effectués qu'à partir de 12 points car il est quasiment impossible de trouver deux individus présentant 12 points caractéristiques identiques, même dans une population de plusieurs millions de personnes. [12]

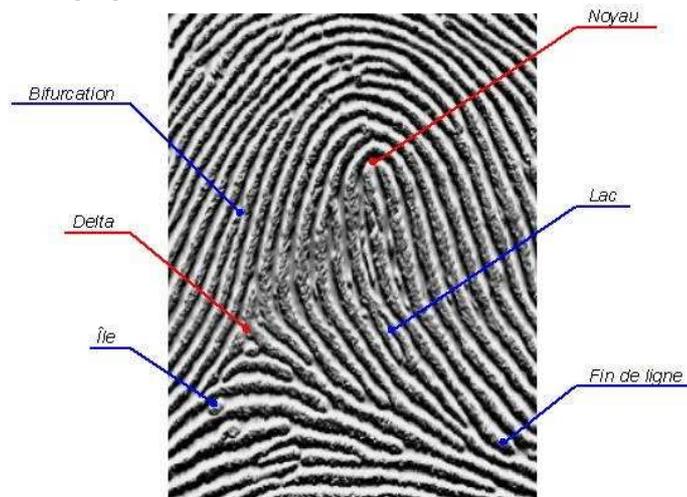


Figure 2-2 : Les différents points caractéristiques d'une empreinte digitale.

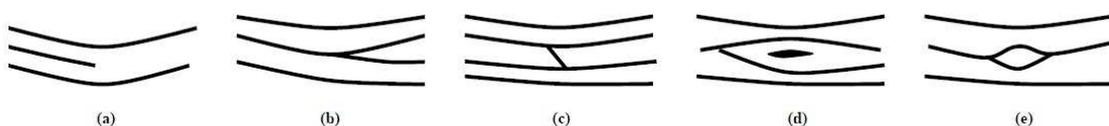


Figure 2-3 : (a) Terminaison, (b) bifurcation, (c) pont, (d) lac et (e) île [8].

La reconnaissance d'empreintes digitales est la technique biométrique la plus ancienne et la plus mature. Cependant les empreintes digitales sont une mesure biométrique assez mal acceptée par les utilisateurs à cause de l'association qui est souvent faite avec la criminologie.

Les appareils d'acquisition numériques des empreintes digitales sont basés généralement sur une technologie :

- **Optique** : L'utilisateur doit placer un ou plusieurs doigts sur une vitre, à travers laquelle l'image recherchée est mise sous éclairage et capturée par une caméra,
- **Capacitive** : Qui effectue l'analyse du champ électrique de l'empreinte digitale pour déterminer sa composition. L'utilisateur place ses doigts directement sur un micro-processeur spécialisé.

Ces capteurs sont souvent doublés d'une mesure visant à vérifier que l'objet photographié est un doigt, par la mesure de sa conductivité, les battements du cœur, la pression sanguine...etc.

Plusieurs méthodes sont employées pour reconnaître les empreintes digitales :

- **Localisation des minuties** : Cette méthode ne retient que l'emplacement des minuties les plus pertinentes. Elle est peu sensible aux déformations des doigts entre plusieurs vérifications (Figure 2-4),
- **Traitement de textures** : Des paramètres issus de certaines propriétés de la texture des empreintes (orientation, fréquence, etc.) sont comparés. Cette méthode permet un traitement très rapide, et donc un temps de réponse très court. [14]



Figure 2-4 : Détection des minuties dans une empreinte digitale.

La qualité principale de cette méthode, est que la caractéristique biométrique est difficile à dupliquer, car l'empreinte est unique et ne se modifie pas au cours du temps, sauf par accident comme une brûlure par exemple. Il faut noter que la probabilité de trouver deux empreintes digitales similaires est pratiquement nulle.

Ce type de système biométrique est utilisé par les institutions financières pour leurs employés et leurs clients. Il se retrouve également dans les magasins de détail, les hôpitaux, les écoles, les aéroports, les cartes d'identité, les passeports, les permis de conduire et de nombreuses autres applications.

## 2.2. L'iris

L'iris (Figure 2-5) est la membrane circulaire, colorée, unique et complexe de l'œil, ses motifs se forment au cours des deux premières années de la vie et sont stables. Cette membrane est constituée d'un réseau de tubes fins dont le diamètre est inférieur à celui d'un cheveu. La forme de l'iris, c'est-à-dire l'enchevêtrement des tubes ne varie que très peu durant la vie de l'individu. [14]



Figure 2-5 : L'iris.

La reconnaissance de l'iris est une technologie récente puisqu'elle ne s'est véritablement développée que dans les années 80 [14]. Pour utiliser cette méthode, une personne doit se placer à une distance qui n'excède pas 40 centimètres de la caméra, cette dernière qui est proche des infrarouges (de rayonnement uniforme) photographie, en évitant les reflets, une tranche de l'iris pour relever les caractéristiques particulières du relief (sillons de contraction, anneaux...). On peut distinguer jusqu'à 244 points de comparaison. Le taux d'erreur des produits disponibles sur le marché est quasiment nul. De plus, l'iris ne se modifie presque pas avec le vieillissement.

Cette méthode peut être améliorée en utilisant les caractéristiques dynamiques de l'œil : réactivité de la pupille (dilatation/rétraction) par rapport à la quantité de lumière, étude de l'iris dans l'infrarouge et l'ultraviolet.

La qualité principale de cette méthode est la fiabilité élevée; elle est donc aussi considérée comme une des méthodes biométriques les plus fiables, car l'iris n'est pour l'instant pas modifiable, même pas par intervention chirurgicale (de plus le taux d'erreur avoisine le 0%). Malgré cela, il existe quelques limitations empêchant la généralisation de cette méthode biométrique :

- **Difficulté de l'extraction** : l'iris est derrière le cornet qui est un miroir réfléchissant les cils,
- **Non acceptée** : Bien que la reconnaissance de l'iris soit moins contraignante que la reconnaissance de la rétine, les gens ont également du mal à accepter cette biométrie,
- **Taille Variable de l'iris** : Suivant la luminosité de l'environnement ou la fatigue,
- **Coût élevé** : l'appareil de mesure est très coûteux. [16]

La reconnaissance de l'iris est utilisée dans le secteur financier pour les employés et les clients, dans les guichets automatiques, pour le paiement dans les supermarchés, dans les institutions carcérales, dans les hôpitaux et dans les aéroports.

### 2.3. Rétine

La rétine (Figure 2-6) est la paroi interne et opposée de l'œil sur laquelle se projettent les images que nous voyons. Cette paroi est tapissée par un réseau de vaisseaux sanguins, qui forment un motif unique pour chaque individu.

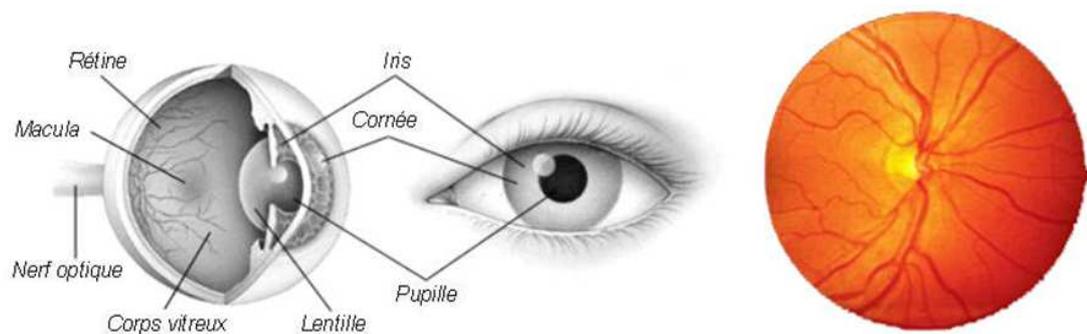


Figure 2-6 : (a) Emplacement de la rétine dans l'œil humain (b) Rétine humaine [14]

Les motifs formés par les veines sous la surface de la rétine sont uniques et stables dans le temps. Ils ne peuvent être affectés que par certaines maladies. Pour ces raisons, la reconnaissance de la rétine est actuellement considérée comme une des méthodes biométriques les plus sûres.

L'image est obtenue en projetant sur l'œil un rayon lumineux de faible intensité dans les fréquences visibles ou infrarouges. L'œil doit être situé très près de la tête de lecture et l'utilisateur doit fixer son regard sur un point déterminé pendant plusieurs secondes ce qui demande une grande coopération de sa part. À ce moment, un faisceau lumineux traverse l'œil jusqu'aux vaisseaux sanguins capillaires de la rétine. Le système localise et capture ainsi environ 400 points de référence. Les

personnes hésitent en général à approcher un organe aussi sensible que l'œil près de l'appareil de mesure ce qui explique pourquoi cette méthode est mal acceptée par le grand public. [14]

Les qualités de cette méthode sont :

- Une fiabilité élevée : Un taux d'erreur presque nul,
- Risques de fraudes quasiment nuls : Puisque la caractéristique biométrique est interne.

Ses inconvénients sont :

- Pas très acceptable : Cette méthode n'est pas très acceptable par les utilisateurs car elle doit se faire à très faible distance du capteur,
- Une forte alcoolémie ou un diabète modifie le réseau veineux rétinien,
- Coût élevé : l'appareil de mesure est très coûteux.

Même si cette technologie est l'une des plus sûres au monde, elle n'est pas facile à mettre en œuvre, et pas très agréable pour les usagers. Du coup, son utilisation reste limitée.

## 2.4. La Voix

---

La reconnaissance du locuteur vise à déterminer les caractéristiques uniques de la voix de chaque individu. Bien que généralement classée comme caractéristique comportementale, la voix se trouve à la frontière de cette dernière et peut aussi être classée comme caractéristiques physiques. En effet, une grande partie de cette caractéristique est déterminée par le conduit vocal ainsi que les cavités buccales et nasales. La voix n'est pas un attribut permanent. Elle change bien entendu avec l'âge mais peut être aussi affectée temporairement par l'état de santé ou émotionnel du locuteur. Cette biométrie est en général très bien acceptée car la voix est un signal naturel à produire.

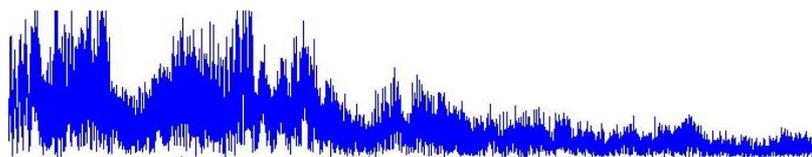


Figure 2-7 : Spectre d'un signal vocal.

Initialement, une table de référence de la voix d'une personne doit être construite. Pour ce faire, celle-ci doit lire une série de phrases ou de mots à plusieurs reprises. Plusieurs caractéristiques de la voix sont alors extraites comme le débit, la force, la dynamique et la forme des ondes produites. Un individu ne parle pas toujours de la même manière, ce qui nécessite l'application d'une méthode permettant d'éliminer certaines de ces variations. Ces caractéristiques formant une empreinte unique sont ensuite traitées par un algorithme et conservées pour comparaison ultérieure. Il existe différentes modalités de reconnaissance du locuteur :

- **Texte-fixe** : le modèle du locuteur est entraîné avec un texte (un mot, une phrase) qui sera aussi utilisé au moment du test. C'est une manière indirecte de combiner reconnaissance de la voix et sécurité basée sur une connaissance car, tant qu'un imposteur ne connaît pas le mot de passe de l'utilisateur, ses chances de tromper le système sont extrêmement faibles. Cependant, si un imposteur enregistre un utilisateur, il lui suffit ensuite de repasser l'enregistrement pour tromper le système,
- **Texte Aléatoire** : ce mode évite l'inconvénient précédent, pendant la phase de reconnaissance, le système prompte aléatoirement une ou plusieurs phrases que le locuteur doit répéter.
- **Indépendant du texte** : le locuteur peut parler librement. Ce mode est utile lorsque l'on veut reconnaître un locuteur sans sa coopération. [8]

Cette méthode est la seule à permettre une identification à distance et sans visibilité de prêt elle peut être effectuée par exemple à l'aide d'un appel téléphonique. Malgré cela, cette méthode a ses limitations : elle ne peut pas être placée dans un endroit public.

Elle est utilisée dans des secteurs comme les centres d'appel, les opérations bancaires, l'accès à des comptes, sur PC domestiques, pour l'accès à un réseau ou encore pour des applications judiciaires.

## 2.5. Signature

Les systèmes de reconnaissance de l'écriture analysent les caractéristiques spécifiques d'une signature comme la vitesse, la pression sur le crayon, le mouvement, les points et les intervalles de temps où le crayon est levé. L'utilisateur de cette technologie signe généralement avec un stylo électronique sur une tablette graphique. Ces données sont enregistrées pour comparaison ultérieure. Il existe deux modes de reconnaissance :

- **Statique** : Ce mode n'utilise que l'information géométrique de la signature,
- **Dynamique** : Ce mode utilise à la fois l'information géométrique et dynamique, c'est-à-dire les mesures de vitesse, d'accélération... il est plus riche en informations que le mode statique et donc plus discriminant. De plus, si un imposteur veut dupliquer une signature à partir d'un exemple, il n'a pas accès à l'information dynamique. [14]

La signature a l'avantage par rapport aux autres mesures biométriques d'être couramment utilisée pour les transactions. Pour cette raison, la signature est en général bien acceptée comme moyen d'identification. Le problème de la reconnaissance par signature provient de la très grande variabilité qui existe entre deux occurrences de la signature d'un même individu. De plus, la signature peut être affectée par l'état de santé ou émotionnel de l'individu.

## 2.6. Formes de la main

Cette méthode consiste à déterminer les caractéristiques de la main d'un individu : la forme, la longueur, la largeur, la courbure des doigts, etc. En tout, 90 points caractéristiques sont nécessaires pour identifier une personne. Les systèmes de reconnaissance de la géométrie de la main sont simples d'usage. L'utilisateur doit poser la paume de sa main sur une plaque qui possède des guides afin de l'aider à positionner ses doigts (Figure 2-8). Ces appareils peuvent être difficiles à utiliser pour certaines catégories de populations pour lesquelles étendre la main est un problème, telles que les personnes âgées. Une photo de la face de la main est ensuite prise par un appareil photo numérique. Une photo de profil peut aussi être prise pour obtenir de l'information sur l'épaisseur de la main. En raison de la taille du système de capture, ce type de technologie est limité à certaines applications.

La géométrie de la main a un faible pouvoir discriminant et les systèmes peuvent être facilement trompés par de vrais jumeaux ou même par des personnes de la même famille. Il existe une alternative à la géométrie de la main : la géométrie des doigts qui s'appuie sur la forme du majeur et de l'index. [8]

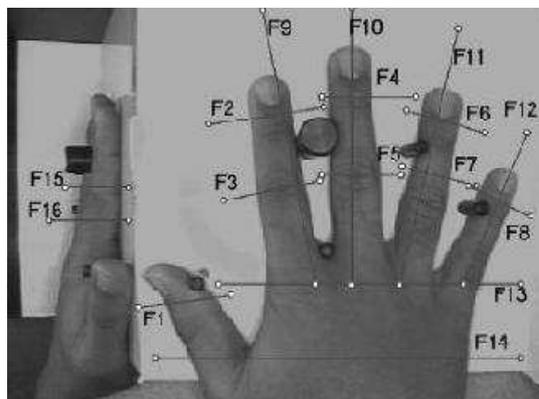


Figure 2-8 : Méthode biométrique, géométrie de la main [17].

L'identification biométrique de la main revêt deux formes :

- L'empreinte palmaire, analogue à l'empreinte digitale,
- L'image tridimensionnelle de la main: quelques traits caractéristiques sont gardés en mémoire comme : la taille et la largeur des doigts, l'espace entre les différentes parties de la main.

Il faut noter que, comme pour les empreintes digitales, on vérifie la validité de la mesure par la conductivité ou la température de la main. On peut aussi l'associer à l'empreinte du réseau veineux.

Cette technologie est utilisée pour contrôler l'accès à des zones sensibles et où un grand nombre de personnes circulent comme lors de Jeux Olympiques, aux frontières, dans les aéroports et dans les grands parcs d'attractions.

### 2.7. Dynamique de frappes

---

Il s'agit d'une technique de reconnaissance de personnes basée sur le rythme de frappe qui leur est propre. C'est une solution purement logicielle. Elle est appliquée au mot de passe qui devient ainsi beaucoup plus difficile à « imiter ».

Lors de la mise en place de cette technique, il est demandé à l'utilisateur de saisir son mot de passe une dizaine de fois de suite. A l'aide d'un algorithme qui exploite le temps d'appui sur chaque touche et le temps entre chaque touche, la dizaine de saisie est « moyenné » pour bâtir un Profil de frappe de l'utilisateur qui servira de référence aux accès suivants, en suivant la même approche, la saisie du mot de passe sera couplée à un profil de frappes qui sera comparé au profil de référence. Le droit d'accès est alors accordé en fonction du niveau de ressemblance de ce profil avec la référence. Suivant le degré de filtrage qu'un administrateur aura défini, cet accès sera plus ou moins difficile.

Cette méthode peut être gênante à l'utilisateur dans le cas d'un changement de claviers (AZERTY, QWERTY... etc). [14]

### 2.8. Reconnaissance de visages

---

Plusieurs méthodes de reconnaissance de visages ont été proposées durant ces 30 dernières années, suivant deux grands aspects :

- La reconnaissance à partir d'images fixes,
- Et la reconnaissance à partir de séquences d'images (ou vidéo).

La reconnaissance de visages basée sur la vidéo est préférable à celle basée sur des images fixes, puisque l'utilisation simultanée des informations temporelles et spatiales aide dans la reconnaissance.

Le visage est certainement la caractéristique biométrique que les humains utilisent le plus naturellement pour s'identifier entre eux, ce qui peut expliquer pourquoi elle est en général très bien acceptée par les utilisateurs. Le système d'acquisition est soit un appareil photo, soit une caméra numérique. La difficulté de la reconnaissance de visage varie énormément suivant que l'acquisition se fait dans un environnement contrôlé ou non :

- Dans un environnement contrôlé, des paramètres tels que l'arrière-plan, la direction et l'intensité des sources lumineuses, l'angle de la prise de vue, la distance de la caméra au sujet sont des paramètres maîtrisés par le système,
- Dans un environnement non-contrôlé, une série de prétraitements sont souvent nécessaires avant de faire la reconnaissance. Il faut tout d'abord détecter la présence ou l'absence de visage dans l'image (Face detection). Le visage doit ensuite être segmenté (Face segmentation). Enfin, si nous travaillons sur un flux vidéo, le système doit suivre le visage d'une image à l'autre (Face tracking).

### 2.8.1. Le processus de reconnaissance de visage

Tout processus automatique de reconnaissance de visages doit prendre en compte plusieurs facteurs qui contribuent à la complexité de sa tâche, car le visage est une entité dynamique qui change constamment sous l'influence de plusieurs facteurs.

Dans le monde physique, il y a trois paramètres à considérer : l'éclairage, la variation de posture et l'échelle. **La variation de l'un de ces trois paramètres peut conduire à une distance entre deux images du même individu, supérieure à celle séparant deux images de deux individus différents.**

Généralement le processus de reconnaissance de visage comporte les étapes suivantes :

- **Le codage** : consiste en l'acquisition d'image et sa digitalisation, il comporte un risque de bruit et donne lieu à une représentation 2-D (la matrice des niveaux de gris) pour un objet 3-D (le visage),
- **Le prétraitement** : il faut éliminer le bruit par des techniques de traitement et de restauration d'images et procéder à une détection de visages, cette opération est très complexe, surtout dans le cas où l'image contient plusieurs visages ou dans le cas où l'arrière-plan n'est pas neutre. La restauration d'images ou l'élimination du bruit consiste à compenser les dégradations connues ou estimées et rétablir une bonne qualité de l'image,
- **L'analyse** : Appelée aussi indexation, représentation, modélisation ou extraction de caractéristiques, il faut extraire de l'image les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de décision. Le choix de ces informations utiles revient à établir un modèle pour le visage, elles doivent être discriminantes et non redondantes,
- **L'apprentissage** : consiste à mémoriser les représentations calculées dans la phase analyse pour les individus connus qui constituent la BDD, généralement les deux étapes d'analyse et d'apprentissage sont confondues et regroupées en une seule étape,
- **La décision** : pour estimer la différence entre deux images, il faut introduire une mesure de similarité comme par exemple la distance euclidienne.

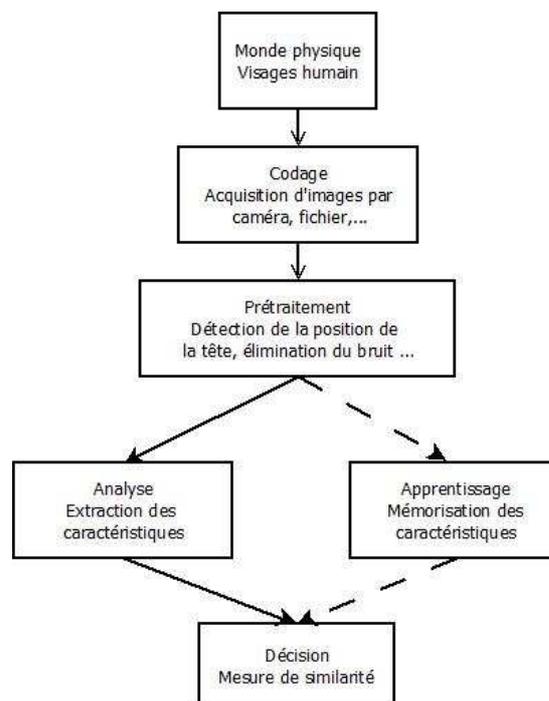


Figure 2-9 : Le Schéma général d'un système de reconnaissance de visage [18].

### 2.8.2. Les classes des techniques de reconnaissance de visages

On peut répartir l'ensemble des techniques de reconnaissance de visages basées sur les images fixes, en trois grandes catégories : les méthodes globales, les méthodes locales et les méthodes hybrides.

#### 2.8.2.1. Méthodes locales

Ce sont des méthodes géométriques, appelées aussi les méthodes à traits, à caractéristiques locales, ou analytiques.

L'analyse du visage humain est donnée par la description individuelle de ses parties et de leurs relations. Ce modèle correspond à la manière avec laquelle l'être humain perçoit le visage, c'est-à-dire, à nos notions de traits de visage et de parties comme les yeux, le nez, la bouche, etc. la plupart des travaux réalisés se sont concentrés sur l'extraction des traits à partir d'une image de visage et sur la définition d'un modèle adéquat pour représenter ce visage.

Un certain nombre de stratégies automatiques et semi-automatiques ont modélisé et classé les visages sur la base de distances normalisées et angles entre points caractéristiques. La stratégie est qualifiée d'automatique ou non selon que la phase d'extraction des points est faite par l'ordinateur ou qu'elle est assistée par un opérateur. Cette phase constitue l'étape clé du processus car la performance du système entier dépend de la précision avec laquelle les informations utiles sont extraites.

L'avantage de ces méthodes réside dans la prise en compte de la particularité du visage en tant que forme naturelle à reconnaître, en exploitant les résultats de la recherche en neuropsychologie et psychologie cognitive sur le système visuel humain. La difficulté éprouvée lors de la considération de plusieurs vues du visage, ainsi que le manque de précision dans la phase « extraction » des points, constituent leur inconvénient majeur. En plus, ces techniques sont facilement affectées par l'information non pertinente.

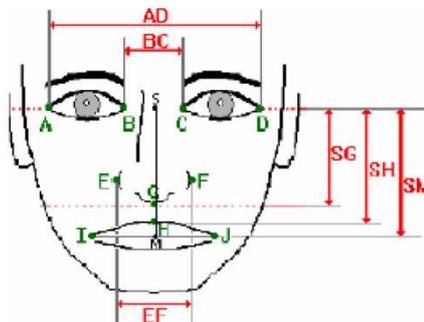


Figure 2-10 : Distance entre points caractéristiques pour la méthode locale [19].

#### 2.8.2.2. Méthodes globales

Les méthodes globales sont des méthodes utilisées dans la reconnaissance de visage, ces méthodes utilisent la région entière du visage comme entrée à l'algorithme de reconnaissance. Ce sont des techniques très réussies et bien étudiées. Ces méthodes offrent les meilleures performances, mais le problème de stockage des informations extraites lors de la phase « apprentissage » reste l'inconvénient majeur.

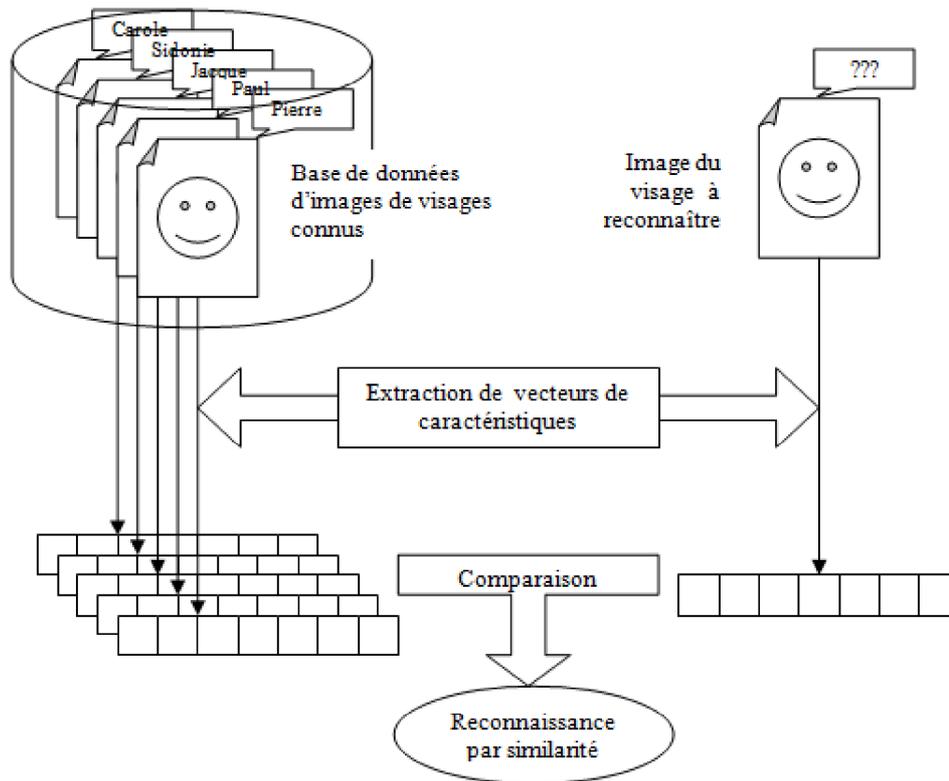


Figure 2-11 : Principe général d'un système de reconnaissance de visages par méthode globale.

### 2.8.2.3. Méthodes hybrides :

Elles combinent les deux types de méthodes, offrant ainsi potentiellement le meilleur des deux types de méthodes. Elles se basent sur le fait que le système de perception humain emploie les caractéristiques locales et la région entière du visage pour l'identification. [18]

## 3. Comparaison entre les différentes méthodes biométriques :

Comme déjà cité, dans le premier chapitre, les critères principaux de comparaison entre les différentes méthodes biométriques sont :

- **Le coût** : un coût faible est souhaité,
- **Effort** : La gêne causée par la méthode aux utilisateurs, une gêne minimale est souhaitée,
- **Précision** : Le taux en pourcentage du nombre de résultats corrects sur la totalité des essais, une bonne précision est souhaitée,
- **Intrusion** : La possibilité qu'un imposteur passe par le système biométrique, une faible intrusion est souhaitée.

Voici à titre indicatif le résultat d'une étude effectuée par une compagnie Américaine, l'International Biometric Group (à New York based integration and consulting firm), présentant les différents critères pour chaque type de technique biométrique.

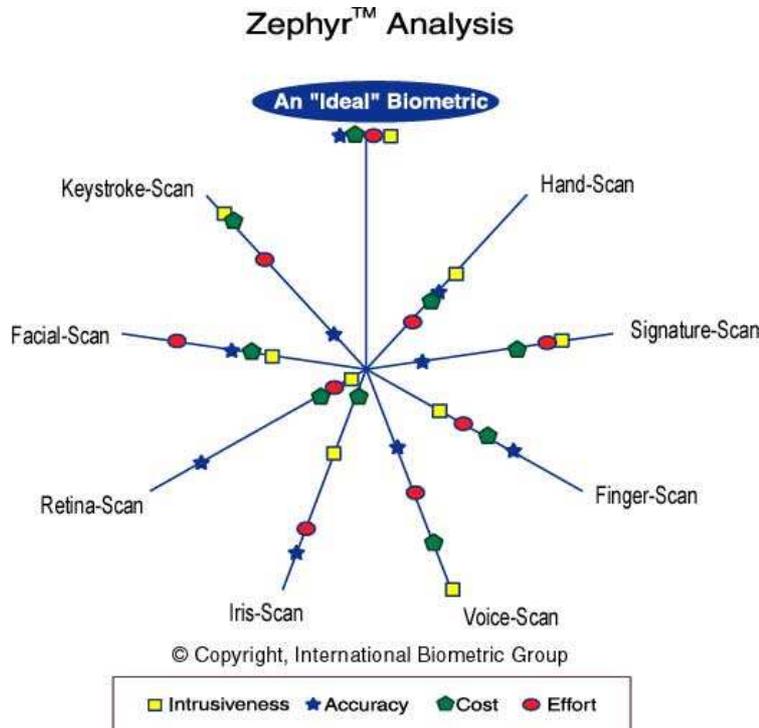


Figure 2-12 : Comparaison Entre les différentes méthodes biométriques.

#### 4. Justification du choix de la reconnaissance de visage

Notre travail de PFE, utilise une méthode globale de reconnaissance de visage en implémentant les algorithmes :

- De Décomposition en Paquets d'Ondelettes (WPD : Wavelet Packet Decomposition) pour la compression des images d'entrée,
- D'Analyse en composante principale (PCA : Principal component Analysis) pour la reconnaissance des visages

En se référant à la figure 2.12, le choix de cette méthode biométrique est basé sur les critères suivants :

- **Une méthode Acceptable** : C'est une méthode qui ne cause pas de gêne aux utilisateurs, donc elle est acceptable,
- **Temps minimal** : l'identification ou la vérification par cette méthode ne nécessite pas un temps important, donc elle peut être utilisée dans des applications temps réel,
- **Faible coût** : Cette solution, n'est pas chère, car seule une simple Webcam comme capture peut être utilisée en combinaison soit avec un PC ou un circuit dédié comme les FPGA.

# Chapitre 3 : Généralités sur la théorie des ondelettes.

## 1. Introduction [19]

Il est connu qu'un signal peut être toujours exprimé comme étant la somme de « sin » et « cos » de fréquences différentes, cette somme est connue sous le nom de « décomposition en série de Fourier ». L'inconvénient majeur de la TF d'un signal  $x(t)$  est la présence de la résolution fréquentielle seulement, c-à-d que malgré notre connaissance des différentes fréquences présentes dans notre signal on ignore leurs emplacements. Pour remédier à ce problème plusieurs solutions ont été proposées afin d'obtenir une représentation temps-fréquence du signal.

Une des solutions proposées pour représenter le signal dans un espace temps-fréquence est la décomposition du signal  $x(t)$  à l'aide d'une fenêtre  $w_0(t)$  en plusieurs morceaux, ensuite analyser chaque morceau par la TF, cela va donner une bonne information sur l'instant et la durée de présence d'une fréquence  $f$ , mais le choix de la fenêtre  $w_0(t)$  n'est pas facile.

Un autre problème dans l'utilisation des fenêtres est que dans le domaine fréquentiel la convolution du signal  $x(t)$  avec la fenêtre  $w_0(t)$  est considérée comme une multiplication, on aura donc un étalement des composantes fréquentielles du signal sur l'axe des fréquences. Ce qui induit une perte dans la résolution fréquentielle.

Le principe décrit ci-dessus est celui d'incertitude d'« Heisenberg » qui déclare qu'en traitement de signal, il est impossible de savoir exactement la fréquence et l'instant où cette fréquence est apparue, en d'autres termes un signal ne peut être représenté par un simple point dans l'espace temps-fréquence.

La transformée en ondelettes est la solution la plus récente qui donne une représentation temps-fréquence du signal  $x(t)$  que la TF ne peut fournir. Inventée par Jean Morlet, un ingénieur français dans les années 80.

L'utilisation d'une fenêtre  $w_0(t)$  qui peut être traduite, dilatée ou compressée résout le problème de décomposition du signal. La fenêtre translate le long du signal et à chaque position le spectre du signal est calculé, ensuite ce procédé est répété plusieurs fois avec une fenêtre soit dilatée ou compressée par rapport à  $w_0(t)$ . A la fin on obtient un ensemble de spectres de notre signal chacun avec une résolution différente (taille de la fenêtre utilisée  $w_0(t)$ ) c'est pour cela que l'on parle aussi d'une analyse multi-résolutions dans la littérature.

La Figure 3-1 représente la forme d'une onde et une ondelette et la Figure 3-2 donne le page de l'espace-temps-fréquence pour la transformée de Fourier à fenêtre et pour l'analyse en ondelettes.

Le choix du terme ondelettes vient du fait que les fonctions ainsi définies ressemblent à de petites ondes, comme on peut le constater sur la Figure 3-1. Le terme anglais est « wavelet » qui signifie petites vagues.

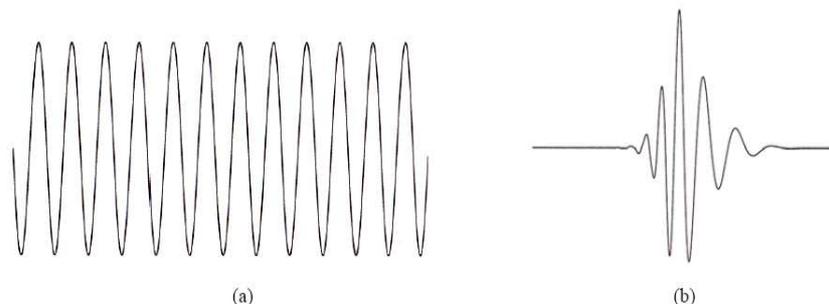


Figure 3-1 : (a) Onde (b) Ondelette. [30]

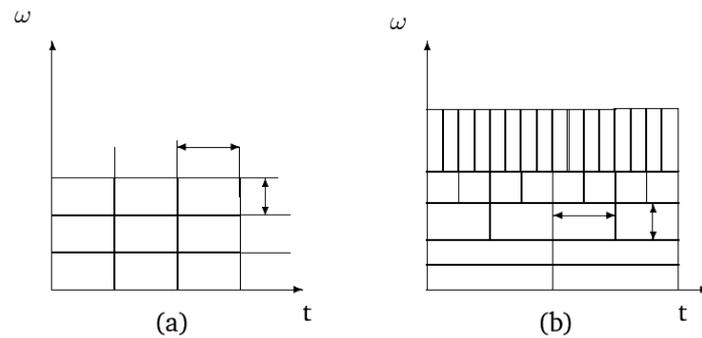


Figure 3-2 : Pavage du plan temps-fréquence. (a) Transformée de Fourier, (b) transformée en ondelettes. [26]

Souvent lorsqu'on parle d'ondelette on n'utilise pas le terme temps-fréquences mais le terme temps-échelle où l'échelle est définie comme étant l'inverse de la fréquence. Et avec le terme temps-échelle, on utilise souvent les termes grandes et petites échelles qui signifient :

- **Grande échelle** : est le signal d'une façon générale (The big picture), l'équivalent des basses fréquences,
- **Petite échelle** : c'est pour les détails de notre signal, l'équivalent des hautes fréquences.

Le passage des grandes échelles vers les plus petites sera comme un « zoom in » sur le signal.

### 1.1. Définition d'une ondelette [22]

Une ondelette est une fonction  $\psi(t)$  de  $L^2(\mathbb{R})$  (espace de Hilbert) ayant des propriétés de localisation et d'oscillation. La localisation exprime le fait que  $\psi(t)$  est à décroissance rapide à l'infini. L'énergie d'une ondelette est contenue dans un intervalle fini. Idéalement, l'ondelette est une fonction nulle en dehors d'un intervalle fini. Autrement dit, c'est une fonction à support compact. Elle est de nature oscillante, et son moment d'ordre « 0 » est nul soit :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (3-1)$$

Le caractère oscillant et la localisation signifient que la transformée de Fourier de l'ondelette est localisée, c'est à dire que l'ondelette a un contenu fréquentiel limité à une certaine bande de fréquences. La Figure 3-3 illustre la représentation fréquentielle d'une ondelette mère et celle de ses deux filles générées par dilatation et compression.

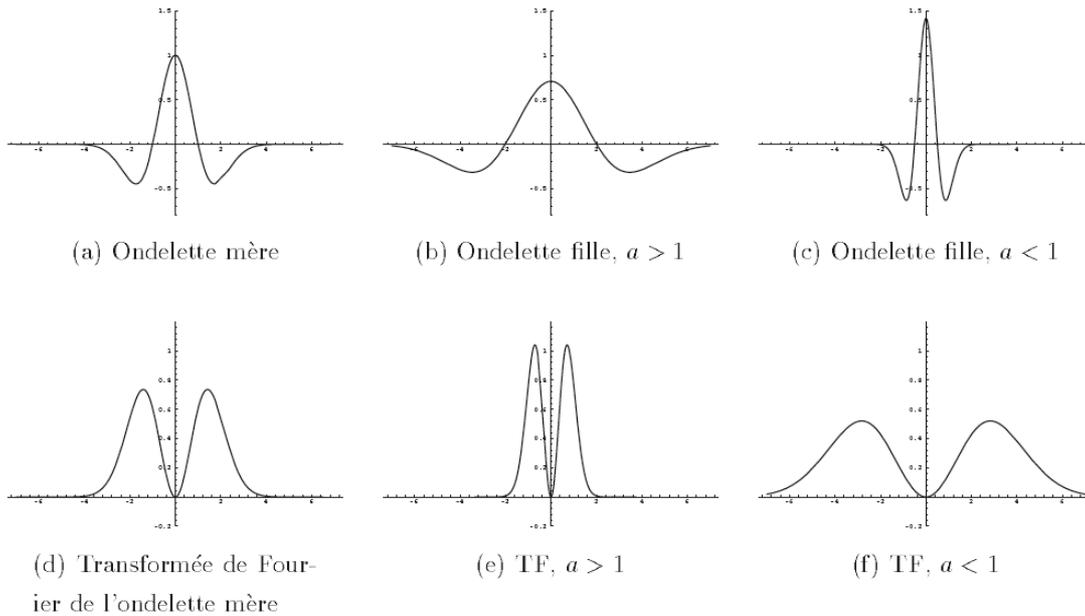


Figure 3-3 : Exemples d'ondelettes unidimensionnelles et leur transformée de Fourier.

A partir d'une ondelette mère (Mother wavelet), on peut créer une base hilbertienne de fonction de  $L^2(\mathbb{R})$  par translation «  $b$  » et dilatation ou compression «  $a$  » de l'ondelette mère :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right), \quad a \in \mathbb{R}^{+*} \text{ et } b \in \mathbb{R} \quad (3-2)$$

Dans ce chapitre, on présentera d'abord la transformée en ondelette continue et discrète et on verra comment une TO peut être réalisée par filtrage, on parlera ensuite des différentes applications de ce nouvel outil et on finira par décrire la façon dont on a utilisé les ondelettes dans notre PFE.

## 2. La transformée en ondelettes continues [19]

La transformation en ondelettes décrite dans l'introduction est connue sous le nom de transformée en ondelettes continue et peut être définie par la relation suivante :

$$\gamma(s, \tau) = \int f(t) \psi_{s,\tau}(t) dt \quad (3-3)$$

Cette équation montre comment une fonction  $f(t)$  peut être décomposée en une série de fonctions élémentaires  $\psi_{(s,\tau)}(t)$  nommées ondelettes, «  $s$  » et «  $\tau$  » représentent les nouvelles dimensions de notre résultat qui sont : le facteur d'échelle et de translation. L'opération inverse de la transformée en ondelette est donnée par l'équation :

$$f(t) = \iint \gamma(s, \tau) \psi_{s,\tau}(t) d\tau ds \quad (3-4)$$

Les ondelettes filles sont générées à partir d'une ondelette  $\psi(t)$  nommée « ondelette mère », par translation avec le facteur «  $\tau$  » et dilatation ou compression avec le facteur «  $s$  ».

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (3-5)$$

Dans l'équation 3-5 le paramètre «  $s$  » représente le facteur échelle (dilater/compresser en anglais) et le paramètre «  $\tau$  » représente le facteur translation, le «  $s^{-1/2}$  » est pour que les différentes ondelettes générées aient la même énergie. (Voir Figure 3-3)

Il faut noter que dans les équations précédentes les fonctions d'ondelettes ne sont pas spécifiées car il n'y a pas une seule, contrairement à la transformée de Fourier ou à d'autres transformations où la fonction utilisée est soit une *exp, sin, cos, ...*

### 3. La transformée en ondelettes discrètes [19]

Pour rendre la transformée en ondelettes pratique, on doit d'abord la discrétiser car on ne peut pas utiliser l'équation 3-3 pour effectuer la transformation d'ondelettes d'un signal  $x(t)$ , à cause de :

- La redondance de la TO, parce-que dans l'équation 3-3 le calcul de la transformée d'ondelette se fait par translation et mise à l'échelle de l'ondelette mère d'une façon continue et en même temps par calcul de la corrélation entre le signal  $x(t)$  et l'ondelette obtenue. Il est clair que les ondelettes obtenues par variation de l'échelle ou translation sont loin d'être une base orthogonale c'est pour cela que les coefficients obtenus par le calcul de la transformée seront très redondants, et comme dans la plupart des applications il est préférable de supprimer cette redondance,
- Le nombre infini de coefficients qui peut être calculé avec l'équation 3-3 à cause du nombre d'ondelettes infini obtenu à partir de l'ondelette mère,
- Le manque de solution analytique pour l'intégrale de l'équation 3-3. Donc le calcul de l'intégrale ne peut être réalisé que par des méthodes numériques et à l'aide d'ordinateur.

Pour contourner les différents problèmes cités ci-dessus, les ondelettes discrètes ont été introduites. Les ondelettes discrètes possèdent des paramètres de translation «  $\tau$  » et de mise à l'échelle «  $s$  » qui varient d'une façon discrète et non pas continue. Ce qui donne l'équation suivante :

$$\psi_{j,k} = \frac{1}{\sqrt{s_0^j}} \psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right) \quad (3-6)$$

Les  $(j, k)$  sont des entiers et le paramètre  $s_0 > 1$  est le pas de dilatation fixe. Le facteur de translation «  $\tau_0$  » dépend du pas de dilatation «  $s_0$  ». Par discrétisation des paramètres «  $s$  » et «  $\tau$  » notre espace temps-échelle sera lui aussi discrétisé par intervalles. Le plus souvent on choisit «  $s_0 = 2$  » pour obtenir un échantillonnage dyadique sur l'axe de fréquence, c'est un choix aussi très utile pour les ordinateurs. Pour le paramètre de translation, on choisit souvent «  $\tau_0 = 1$  » pour obtenir aussi un échantillonnage dyadique sur l'axe du temps.

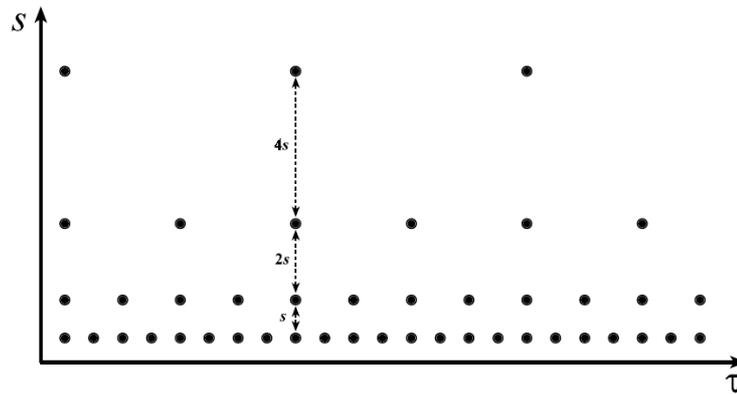


Figure 3-4 : Localisation des ondelettes discrètes dans l'espace temps-échelle sur une grille dyadique.

Lorsque les ondelettes discrètes sont utilisées pour transformer des signaux continus, le résultat sera une série de coefficients, cette transformation est nommée « *décomposition en série d'ondelettes* ». Il faut noter que l'opération inverse qui constitue à retrouver le signal original  $x(t)$  à partir des coefficients obtenus est toujours possible si la condition suivante est vérifiée :

**« L'énergie totale des coefficients d'ondelettes doit être entre deux nombres positifs »**

$$A\|x\|^2 \leq \sum_{j,k} |(x, \psi_{j,k})|^2 \leq B\|x\|^2 \quad (3-7)$$

où  $\|x\|^2$  est l'énergie de  $x(t)$  et  $A > 0, B < \infty$  la valeur de A et B est indépendante de  $x(t)$ .

L'utilisation d'une base orthonormée d'ondelettes est préférable pour faciliter la décomposition et aussi la reconstitution de notre signal  $x(t)$ , la base orthonormée est obtenue par l'utilisation d'ondelettes discrètes, ces dernières peuvent être orthogonales à leurs versions dilatées et translatées aussi par un choix spécial de l'ondelette mère, ce qui veut dire :

$$\int \psi_{j,k}(t) \psi_{m,n}^*(t) dt = \begin{cases} 1 & \text{if } (j = m) \text{ and } (k = n) \\ 0 & \text{ailleurs} \end{cases} \quad (3-8)$$

Avec une base orthonormée la reconstruction du signal  $x(t)$  est la sommation des fonctions d'ondelettes de bases utilisées pour la décomposition pondérée avec les différents coefficients d'ondelettes obtenus :

$$x(t) = \sum_{j,k} \gamma(j,k) \psi_{j,k}(t) \quad (3-9)$$

#### 4. Quelques familles d'ondelettes

Il existe plusieurs fonctions de base qui peuvent être considérées comme des fonctions d'ondelette mère pour la transformée d'ondelettes. Afin d'utiliser au mieux la transformée en ondelette il faut d'abord bien étudier le signal à analyser, ensuite choisir suivant ses caractéristiques l'ondelette mère qui convient le plus.

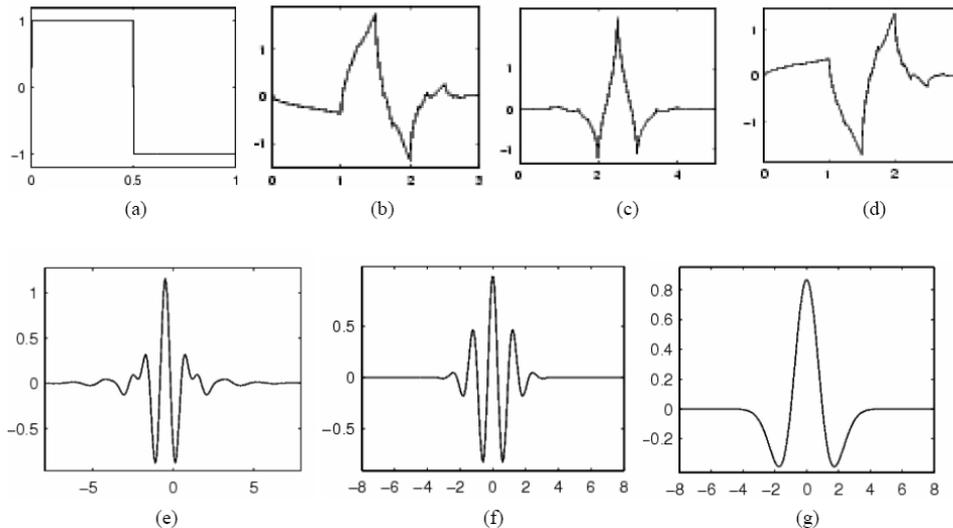


Figure 3-5 : Familles d'ondelettes (a) Haar (b) Daubechies (c) Coiflet (d) Symlet2 (e) Meyer (f) Morlet (g) Mexican Hat. [28]

La Figure 3-5 ci-dessus illustre quelques unes des ondelettes les plus utilisées. L'ondelette de « Haar » est une des plus anciennes et simples ondelettes, elle est aussi la plus utilisée pour la compression d'image, chose que l'on verra par la suite. L'ondelette de « Daubechies » est aussi très populaire et utilisée dans plusieurs applications, elle est nommée ondelette à maximum plat « Maxflat wavelet » parce que son spectre est plat dans les fréquences égales à 0 et  $\pi$ .

## 5. Les filtres passe-bandes et les ondelettes [19]

Malgré l'utilisation des fonctions d'ondelettes discrètes le calcul de la transformée nécessite toujours pour un signal continu un nombre infini d'ondelettes translatées et dilatées. La solution évidente pour ne pas utiliser un nombre infini d'ondelettes est d'utiliser un nombre fini, chose qui pose problème concernant la qualité des résultats obtenus par la suite.

La translation de l'ondelette est bien sûr limitée par la durée du signal analysé, le seul problème restant est celui concernant le facteur d'échelle «  $s$  ». La question qui se pose est la suivante : De combien d'échelles a-t-on besoin pour bien analyser le signal ?

Si on voit la transformée de Fourier d'une ondelette on peut dire qu'elle ressemble à un filtre passe bande (Figure 3-3), de la théorie de transformée de Fourier découle que la compression dans le domaine temporel est équivalent à une dilatation dans le domaine fréquentiel et aussi à une diminution de l'amplitude de notre spectre.

$$x(t) \rightarrow X(f) \quad (3-10)$$

$$x(at) \rightarrow \frac{1}{|a|} X\left(\frac{f}{a}\right) \quad (3-11)$$

Cela veut dire que la compression de l'ondelette mère avec un facteur de deux impliquera une dilatation du spectre de notre fenêtre par deux et aussi une division par deux de toutes les amplitudes du spectre. De même si on dilate l'ondelette par un facteur de deux on aura un spectre compressé par un facteur de deux et des amplitudes multipliées par deux. Si le signal  $x(t)$  est dilaté alors le spectre sera compressé en largeur et son amplitude augmente.

La Figure 3-6 illustre le cas où on applique une dilatation sur un signal  $x(t)$ .

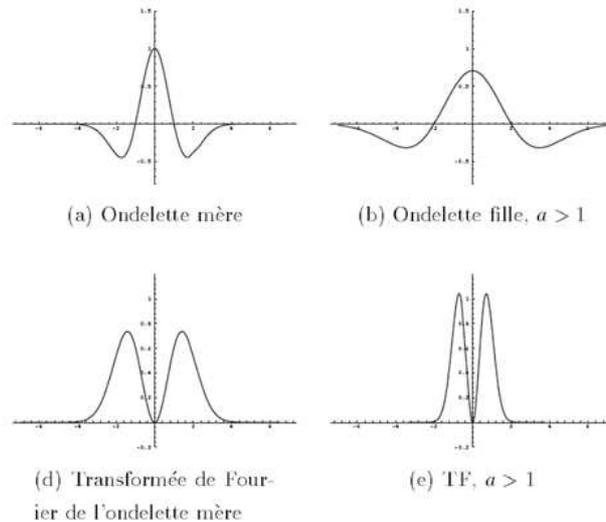


Figure 3-6 : Le spectre du signal  $x(t)$  et de sa version dilatée  $x(at)$ .

En Utilisant cette propriété, on peut couvrir le spectre de notre signal analysé avec les spectres des ondelettes dilatées et compressées. Pour obtenir une bonne couverture du signal analysé les spectres des différentes ondelettes utilisées doivent se toucher comme illustré dans la Figure 3-7.

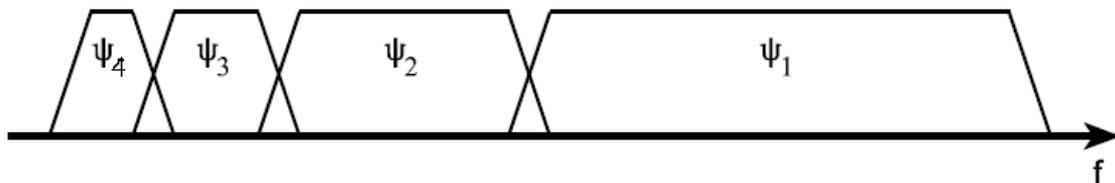


Figure 3-7 : Le spectre des différentes ondelettes résultantes par mise à l'échelle (dilatation) de l'ondelette mère dans le domaine temporel.

Donc si on considère une ondelette comme un filtre passe bande alors une série dilatée d'ondelettes peut être vue comme un ensemble de filtres passe-bandes nommé aussi « band-pass filter bank », si on compare le rapport du centre du spectre sur la largeur du spectre de chaque ondelette on trouve qu'il est le même pour toutes les versions dilatées de l'ondelette mère. Ce rapport est désigné habituellement comme étant « le facteur de fidélité »  $Q$  d'un filtre, dans le cas des ondelettes, on parle de la constante de banque de filtre  $Q$ .

## 6. La fonction d'échelle [19]

Pour effectuer une transformation d'ondelettes avec les filtres passe-bandes, on doit couvrir tout le spectre de notre signal, donc utiliser un nombre infini d'ondelettes car lorsqu'on dilate avec un facteur de deux par exemple notre ondelette dans le domaine temporel, on couvrira seulement la moitié du spectre restant.

La solution choisie pour remédier à ce problème est la suivante : « Lorsqu'on arrive à un niveau où le spectre restant est assez petit, au lieu de continuer à utiliser des filtres passe-bandes on utilise un filtre passe-bas qui couvrira le spectre restant ». Le spectre passe-bas utilisé appartient à la fonction nommée « fonction d'échelle ». La fonction d'échelle a été introduite par **Mallat**, et à cause de la nature passe-bas du spectre de la fonction d'échelle elle est parfois appelée le « filtre moyenneur ».

Si on considère la fonction d'échelle comme un signal simple ce dernier peut être décomposé par transformation d'ondelettes, on écrira donc :

$$\varphi(t) = \sum_{j,k} \gamma(j,k) \psi_{j,k}(t) \quad (3-12)$$

Comme le spectre de la fonction  $\varphi(t)$  couvre la partie restante que la première ondelette n'a pas couvert et comme cette fonction peut être considérée comme un ensemble infini d'ondelettes alors pour analyser un signal, il suffit d'utiliser par combinaison la fonction d'échelle  $\varphi(t)$  et aussi la fonction ondelette  $\psi(t)$ , la fonction d'échelle couvre la partie des ondelettes jusqu'à l'échelle «  $j$  » et le reste est couvert par l'ondelette.

L'introduction de la fonction d'échelle permet de passer à un nombre fini d'ondelettes utilisées ; c'est évident que lorsqu'on utilise les fonctions d'échelles on perd de l'information, de notre point de vue mais d'un point de vue signal on ne perd pas d'information car le signal d'origine peut toujours être reconstitué à partir des coefficients obtenus ; mais d'un point de vue ondelette on néglige quelques valeurs possibles du paramètre d'échelle «  $s$  », c'est pour cela que la largeur de la fonction d'échelle est importante dans la conception d'un système de transformation en ondelettes.

Plus le spectre de la fonction d'échelle est réduit plus de coefficients d'ondelettes sont obtenus. Si une ondelette peut être considérée comme un filtre passe-bande et que les fonctions d'échelles sont considérées comme des filtres passe-bas, donc les séries d'ondelettes dilatées et les fonctions d'échelles forment un banc de filtres « filter bank ».

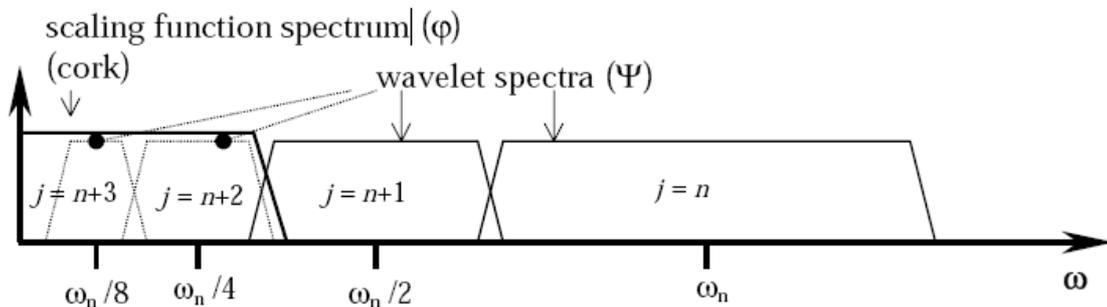


Figure 3-8 : Comment un ensemble infini d'ondelettes est remplacé par une fonction d'échelle.

## 7. Codage en sous-bande [19]

Si nous considérons la transformée en ondelettes comme une banque de filtres, alors nous pouvons envisager que transformer un signal en ondelettes revient à passer ce signal à travers une banque de filtres. Les sorties des différents étages de filtres sont les coefficients d'ondelettes et de fonctions d'échelles.

Analyser un signal en le passant à travers un banc de filtres n'est pas une idée nouvelle mais elle date de plusieurs années sous le nom de codage en sous-bandes (Sub-band coding).

La banque de filtres dont on a besoin dans le codage en sous bandes peut se construire par les deux méthodes suivantes :

- La construction de plusieurs filtres passe-bandes afin de diviser le spectre de fréquence. L'avantage de cette méthode est que la largeur de chaque filtre peut être spécifiée pour mieux analyser les signaux par la suite ; le désavantage est que la procédure de réalisation de tous les filtres prend du temps,

- Par division de spectre du signal en deux parties égales, un passe-bas et un autre passe-haut (low-pass and high-pass). La partie passe-haut contient les petits détails et la partie passe-bas contient une version approchée de notre signal plus quelques détails, que l'on peut extraire par une simple décomposition de la partie passe-bas en deux sous-parties aussi la première passe-bas et la seconde passe-haut et ainsi de suite jusqu'à atteindre le niveau de décomposition désiré. Le processus de la décomposition est illustré dans la Figure 3-9. L'avantage de cette méthode est que l'on a créé seulement deux filtres, le désavantage principal est que le nombre de filtres utilisés pour couvrir le spectre d'un signal est fixe.

**Remarque :**

- L'analyse par codage en sous-bandes est aussi appelée « *analyse multi-résolution* ».
- Avec la méthode des filtres itératifs, on n'a pas besoin d'une fonction d'ondelette mère spécifique à utiliser.

La Figure 3-9 illustre le fractionnement du spectre du signal avec seulement l'utilisation de filtres passe haut et passe bas.

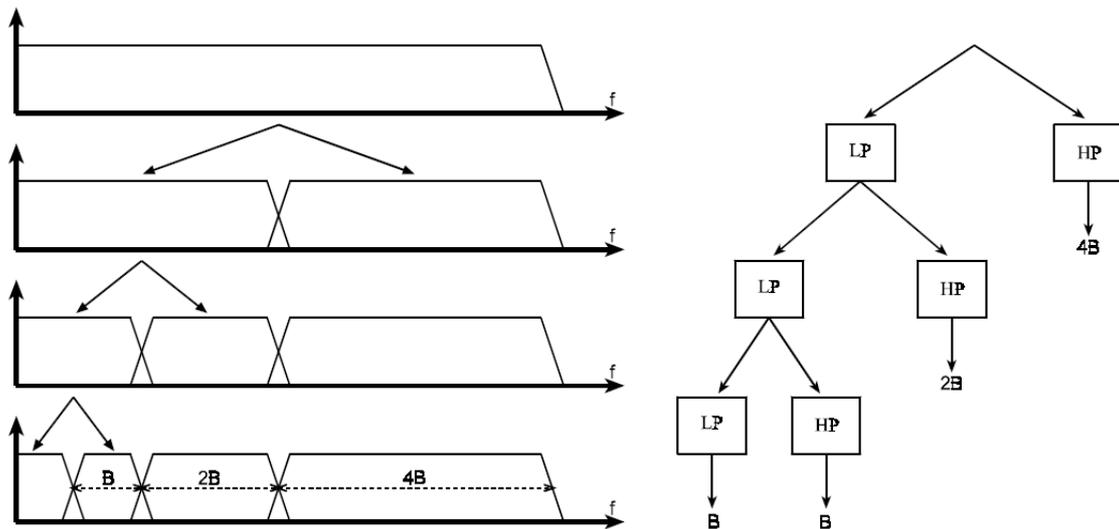


Figure 3-9 : Fractionnement du spectre du signal avec un banc de filtres itéré.

## 8. La transformée en ondelettes discrète

Dans la plupart des applications, les signaux à une ou deux dimensions utilisées sont échantillonnées. On travaille par exemple pour un signal à 1D dans un domaine temporel discrétisé, sachant que la transformée en ondelettes déjà décrite ci-dessus est discrétisée suivant seulement les paramètres translation «  $\tau$  » et le facteur d'échelle «  $s$  » donc pour pouvoir utiliser la TO sur notre signal  $x(nT)$  échantillonné, on doit aussi discrétiser la TO [19].

On peut exprimer une fonction d'échelle  $\varphi(t)$  par un ensemble infini d'ondelettes jusqu'au niveau  $j$ , si on ajoute au spectre de la fonction d'échelle le spectre de l'ondelette on obtiendra une autre fonction d'échelle avec un spectre double du premier donc on peut exprimer la première fonction d'échelle par la seconde parce que toutes les informations dont a besoin sont dans la deuxième. Nous pouvons exprimer cela par la formule dite de multi-résolution [23] ou « two-scale relation » [24]:

$$\varphi(2^j t) = \sum_k h_{j+1}(k) \varphi(2^{j+1} t - k) \tag{3-13}$$

La "two-scale relation" nous informe que la fonction d'échelle à une certaine échelle peut être exprimée par d'autres fonctions d'échelles de niveaux inférieurs translatés (donc petites échelles, ce qui signifie plus de détails).

La première fonction d'échelle remplace un ensemble, de fonctions d'ondelettes donc on peut aussi exprimer ces ondelettes avec des versions translatées de la fonction d'échelle qui sont à l'échelle suivante.

On peut écrire l'ondelette du niveau  $j$  comme suit :

$$\psi(2^j t) = \sum_k g_{j+1}(k) \varphi(2^{j+1} t - k) \quad (3-14)$$

qui est la « two-scale relation » entre la fonction d'échelle et l'ondelette.

Et comme notre signal  $x(t)$  peut être exprimé en utilisant des ondelettes translatées et dilatées jusqu'à une échelle  $j - 1$ , ceci implique que la fonction  $x(t)$  peut être aussi exprimée par des fonctions d'échelles translatées jusqu'à l'échelle  $j$ .

$$x(t) = \sum_k \lambda_j(k) \varphi(2^j t - k) \quad (3-15)$$

Si dans cette équation on utilise  $\varphi(t)$  jusqu'à  $j - 1$  seulement on doit rajouter une ondelette de telle façon à garder le même niveau de détail :

$$x(t) = \sum_k \lambda_{j-1}(k) \varphi(2^{j-1} t - k) + \sum_k \gamma_{j-1}(k) \psi(2^{j-1} t - k) \quad (3-16)$$

Si la fonction d'échelle et d'ondelette sont orthonormées, les différents coefficients s'obtiennent par un simple produit scalaire avec la fonction du signal  $x(t)$ .

$$\lambda_{j-1}(k) = \langle x(t), \varphi_{j,k}(t) \rangle \quad (3-17)$$

$$\gamma_{j-1}(k) = \langle x(t), \psi_{j,k}(t) \rangle \quad (3-18)$$

Si maintenant on remplace  $\varphi_{j,k}(t)$  et  $\psi_{j,k}(t)$  dans les produits scalaires ci-dessus par les équations (3-13) et (3-14) on obtient les résultats suivants :

$$\lambda_{j-1}(k) = \sum_m h(m - 2k) \lambda_j(m) \quad (3-19)$$

$$\gamma_{j-1}(k) = \sum_m g(m - 2k) \lambda_j(m) \quad (3-20)$$

Ces deux équations montrent que les coefficients d'ondelettes et de la fonction d'échelle d'un certain niveau peuvent être calculés par une somme pondérée des coefficients de la fonction d'échelle du niveau précédent. Lorsqu'on fait une TO par l'utilisation d'un banc de filtre on commence toujours par le spectre du signal  $x(t)$  à analyser, donc si on imagine que ce spectre est la sortie d'un filtre passe-bas d'un niveau inférieur alors on peut dire que notre signal échantillonné  $x(kT)$  représente les coefficients de la fonction d'échelle imaginaire donc les éléments  $x(kT)$  seront égaux à  $\lambda(k)$ .

La somme pondérée discrète des coefficients d'ondelette et de la fonction d'échelle est comme celle d'un filtre numérique, puisque nous savons que les coefficients  $\lambda_j(k)$  proviennent de la

partie passe-bas du spectre du signal, les facteurs de pondération  $h(k)$  doivent former un filtre passe-bas. Les coefficients  $\gamma_j(k)$  proviennent de la partie passe-haut du spectre du signal, les facteurs de pondération  $g(k)$  doivent former un filtre passe-haut.

Donc on peut dire que la TO est équivalente à un banc de filtres numériques itéré ce qui nous permettra de parler de la transformées en ondelette discrètes (DWT).

## 9. Applications des ondelettes [20]

Les ondelettes sont un outil récent de traitement du signal permettant d'analyser à plusieurs échelles de temps des propriétés locales de signaux complexes pouvant présenter des zones d'instationnarité. Elles sont à la base de nouvelles techniques d'analyse et de synthèse du signal et s'appliquent pour des problèmes généraux tels que la compression et le débruitage.

En se focalisant sur les applications, on peut dégager trois problèmes généraux pour lesquels les ondelettes se sont révélées très performantes : l'analyse, le débruitage et la compression en particulier la compression d'images.

- **L'analyse** : Une ondelette est une fonction qui oscille comme une onde mais qui est rapidement atténuée. Bien localisée à la fois en temps et en fréquence, elle permet de définir par translation en temps et dilatation en échelle, une famille de fonctions analysantes. Les ondelettes fournissent un cadre pour la décomposition d'un signal sous la forme d'une suite de signaux dits d'approximation de résolution décroissante complétée par une suite de retouches appelées détails,
- **Le débruitage** : Les méthodes de débruitage se basant sur les représentations par ondelettes donnent lieu à des algorithmes très simples, souvent plus performants et faciles à régler que les méthodes traditionnelles grâce à leur adaptativité. Le principe consiste à calculer la transformée en ondelettes des observations, puis à sélectionner, par seuillage, les coefficients à partir desquels un signal est reconstruit par transformation inverse,
- **La compression** : Les ondelettes permettent, grâce à leur représentation généralement très creuse, de réduire considérablement le nombre d'information à coder. Deux applications retentissantes méritent d'être citées ; le stockage de millions d'empreintes digitales par le FBI et la nouvelle norme de compression d'images JPEG 2000. [21]

D'une façon générale tout traitement utilisant les ondelettes peut être résumé par la Figure 3-10 suivante où le traitement peut soit arriver au bout de la chaîne ou s'arrêter au milieu :

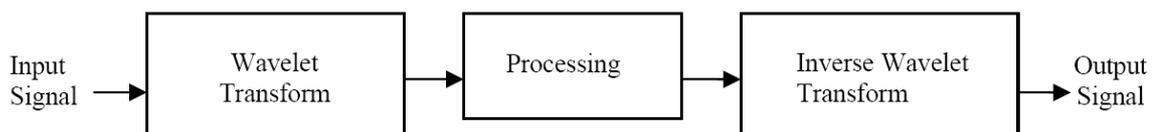


Figure 3-10 : Le traitement de signal en utilisant les ondelettes. [30]

### 9.1. Principe de fonctionnement de la compression DWT

La compression se compose des étapes suivantes :

- Transformations par ondelettes (méthode non destructive et réversible),
- Quantification : Les valeurs de détails du signal inférieures à un certain niveau sont éliminées, cette étape introduit des pertes,
- Codage canal des valeurs restantes.

La méthode de compression par ondelette peut être avec ou sans perte. Dans le cadre de notre PFE nous nous intéressons uniquement à la première étape qui consiste à faire la transformée en ondelettes de l'image et garder seulement la version du signal (image) approchée de résolution inférieure qui est de taille réduite, ce qui nous fera gagner en temps de traitement et en espace mémoire utilisé.

La transformée en ondelettes 2D découle directement de la transformée en ondelette 1D. Ainsi la méthode la plus simple pour obtenir les fonctions d'ondelettes et d'échelle en deux dimensions est de multiplier deux fonctions à une dimension. On obtient donc la fonction d'échelle 2D par multiplication de deux fonctions d'échelle 1D:  $\varphi(x, y) = \varphi(x)\varphi(y)$ . L'obtention des fonctions d'ondelettes est réalisée par multiplication de deux fonctions d'ondelettes 1D ou d'une fonction d'ondelettes avec une fonction d'échelle. Ainsi une transformée en ondelettes 2D possède trois fonctions d'ondelettes :  $\psi_h(x, y) = \varphi(x)\psi(y)$ , cette première fonction d'ondelettes permet de récupérer les détails horizontaux,  $\psi_v(x, y) = \psi(x)\varphi(y)$  pour les détails verticaux,  $\psi_d(x, y) = \psi(x)\psi(y)$  pour les détails diagonaux. Comme la TOD 1D, la TOD 2D peut être implémentée en utilisant un banc de filtres passe-haut et passe-bas.

Suite au passage de l'image à travers ce banc de filtres, on obtient une nouvelle image (Figure 3-11). Cette image est décomposable en quatre parties distinctes, issues du filtrage à travers les quatre filtres de la TOD. Chacune de ces sous-images contient une partie de l'information de l'image de départ. Les blocs LH, HL et HH contiennent tous des détails de l'image, ou informations hautes fréquences. L'image est analysée dans différentes directions; les détails horizontaux sont contenus dans LH, les détails verticaux dans HL et les détails diagonaux dans HH. Le bloc LL contient une image de plus basse résolution (4 fois moins de pixels que l'image initiale) dont les détails dans différentes directions ont été soustraits (principe de la multi-résolution).

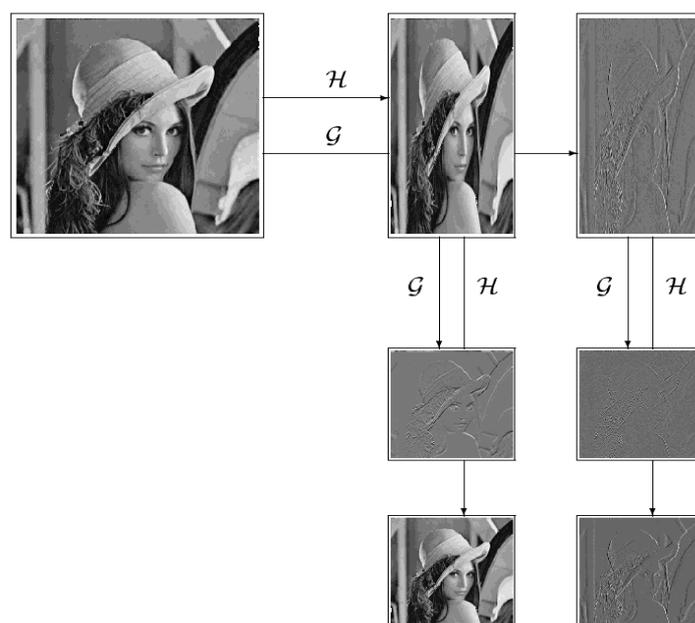


Figure 3-11 : Image décomposée en blocs avec TO. [27]

Le bloc qui sera gardé pour des traitements ultérieurs dans le cadre de notre PFE sera celui *LL* contenant une image de plus basse résolution, la Figure 3-12 suivante illustre ce principe :

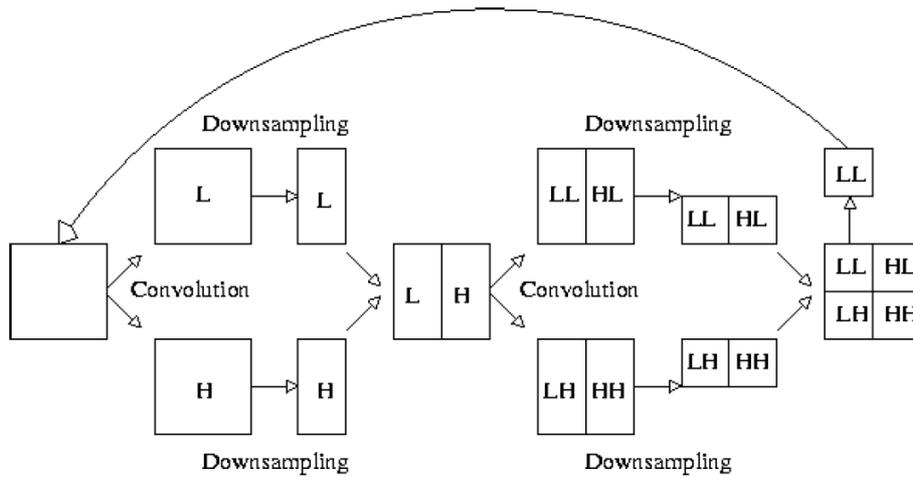


Figure 3-12 : Schéma général de traitement d'une image par ondelettes. [29]

Pour obtenir la TOD de notre image on utilise l'ondelette et la fonction d'échelle de « Haar » pour pouvoir calculer les différentes valeurs de coefficients, la Figure 3-13 illustre les deux fonctions de « Haar » utilisées :

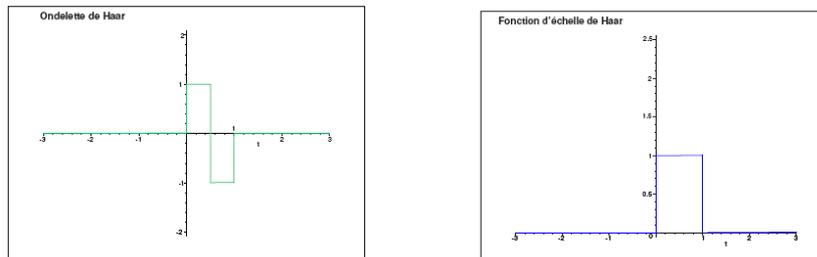


Figure 3-13 : Ondelette et fonction d'échelle de Haar.

La fonction d'échelle de l'ondelette de « Haar » représente le filtre passe bas qui est simplement défini par les coefficients  $\{1/2 ; 1/2\}$ , la sortie de ce filtre représente la moyenne de deux échantillons adjacents du signal.

La fonction d'ondelette de « Haar » qui représente un filtre passe-haut est définie par les coefficients  $\{1/2 ; -1/2\}$  donc la différence de deux échantillons adjacents du signal divisé par deux.

Dans le cas des images, nous n'avons que deux coordonnées selon lesquelles l'échantillonnage est fait ( $x, y$ ), on utilise deux filtres passe-bas et deux passe-haut, une paire pour les  $x = \text{colonnes}$  et l'autre pour les  $y = \text{lignes}$ .

On commence par les colonnes deux à deux et on calcule leur moyenne et différence ensuite on passe aux lignes en procédant de la même façon.

La Figure 3-14 illustre le résultat obtenu dans chaque étape séparément ( $H(x)$  : filtre passe-bas colonnes,  $G(x)$  : filtre passe-haut colonnes,  $H(y)$  : filtre passe-bas lignes,  $G(y)$  : filtre passe-haut lignes).

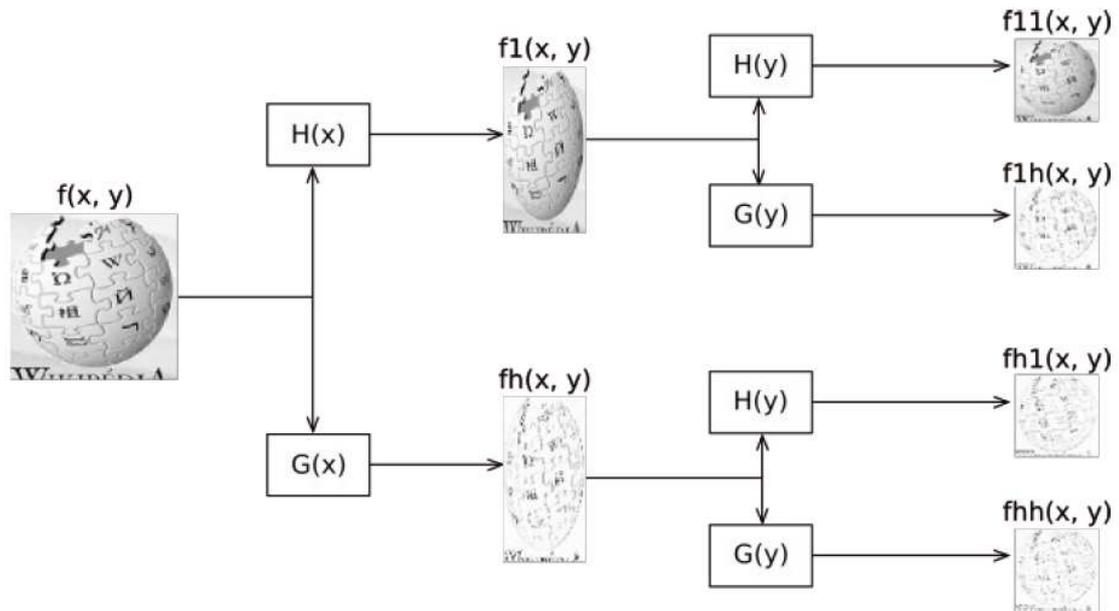


Figure 3-14 : transformée en ondelettes niveau après niveau.

## 10. Conclusion

L'analyse par ondelettes constitue l'une des méthodes numériques temps-échelle et temps-fréquence, pour le traitement du signal et des images des plus utilisées. Elle apporte une aide précieuse dans des problématiques de restauration, de compression, de segmentation et d'analyse.

La conception de décomposition directement dans un espace discret permet d'obtenir un espace de représentation qui possède en général de bonnes propriétés (orthogonalité, stabilité, rapidité de calcul, etc) tout à fait adapté pour le traitement numérique du signal 1D ou 2D. L'orthogonalité permet par exemple d'obtenir des taux de compression importants (absence de redondance d'informations).

La décomposition en ondelettes d'un signal 1D ou 2D (image) nous a permis d'obtenir des versions approchées du signal original présentant les mêmes caractéristiques de ce dernier mais avec l'absence de quelques détails.

# Chapitre 4 : Analyse en Composante Principale (PCA)

## 1. Introduction

L'Analyse en Composantes Principales (ACP) est une méthode de la famille de l'analyse des données et plus généralement de la statistique multivariée, qui consiste à transformer des variables liées entre elles (dites "corrélées" en statistique) en nouvelles variables indépendantes les unes des autres (donc "non corrélées"). Ces nouvelles variables sont nommées "composantes principales", ou axes. Elle permet au praticien de réduire l'information en un nombre de composantes plus limité que le nombre initial de variables [31].

L'algorithme PCA est né des travaux de « M. A. Turk et A. P. Pentland » au « MIT Media Lab », en 1991 [33]. Il est aussi connu sous le nom de « Eigenfaces » car il utilise des vecteurs et des valeurs propres (respectivement Eigenvectors et Eigenvalues en anglais).

L'idée principale consiste à exprimer M images de départ selon une base de vecteurs orthogonaux particuliers - les vecteurs propres - contenant des informations indépendantes d'un vecteur à l'autre. Les données originales seront donc exprimées d'une manière plus appropriée à la reconnaissance du visage.

En théorie de l'information, nous voulons extraire l'information caractéristique d'une image de visage, pour l'encoder aussi efficacement que possible afin de la comparer à une base de données de modèles encodés de manière similaire.

En termes mathématiques, cela revient à trouver les vecteurs propres de la matrice de covariance formée par les différentes images de notre base d'apprentissage [32].

## 2. Rappels Mathématiques [35]

Durant ce sous-chapitre, Les outils mathématiques nécessaires pour l'algorithme PCA seront présentés. Les parties sont indépendantes les unes des autres. Les statistiques sont utilisées quand on veut étudier les relations existantes entre les éléments d'un ensemble de données.

### 2.1. La moyenne

Soit l'ensemble  $X = \{x_1, x_2, \dots, x_N\}$ , la valeur moyenne de cet ensemble est définie par la formule suivante :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i \quad (4-1)$$

Cette valeur nous donne seulement une information sur le milieu (dit aussi centre de gravité) de l'ensemble. La moyenne n'est donc pas une information suffisante, car on peut avoir deux ensembles de même moyenne mais de distributions de valeurs différentes, par exemple :

$$X_1 = \{0, 8, 12, 20\} \text{ Et } X_2 = \{10, 10, 10, 10\} \quad (4-2)$$

### 2.2. L'écart type et la Variance

Les deux ensembles précédents se différencient par la façon dont leurs valeurs varient autour de leurs moyennes, cette variation est appelée « écart type » qui peut être définie aussi par la phrase suivante : « La distance moyenne entre les points de notre ensemble et de leur valeur moyenne », l'équation suivante nous permet de calculer l'écart type :

$$Ecart\ type = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n}} \quad (4-3)$$

En analysant les ensembles X1 et X2 déjà cités, on remarque que l'ensemble X1 a une plus grande variance que X2.

Une autre grandeur très importante dans les statistiques est la « Variance », elle est définie comme « L'écart type élevé au carré », l'équation suivante nous permet de calculer la variance :

$$Variance = \frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n} \quad (4-4)$$

### 2.3. La covariance

La variance n'est calculée que pour des ensembles à une seule dimension. L'objectif principal de l'analyse statistique est de voir s'il y a une relation entre les différentes dimensions. La covariance est la mesure d'une grandeur qui nous permet de savoir comment une variable varie en fonction d'une autre.

Contrairement à la variance, la covariance est calculée entre deux dimensions différentes, la covariance entre la même dimension est une « Variance ». La formule de la covariance est la suivante :

$$Cov(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{n} \quad (4-5)$$

Le signe de la covariance entre deux variables « X » et « Y » indique la façon avec laquelle une dimension varie en fonction de l'autre, c'est-à-dire :

- $Cov(X, Y) > 0$  : « X » varie proportionnellement à la variation de « Y »,
- $Cov(X, Y) < 0$  : « X » varie en inversement proportionnel la variation de « Y »,
- $Cov = 0$  : La variable « X » est indépendante de la variable « Y ».

La covariance est possible dans un espace à dimension élevé, où on pourra calculer cette dernière entre chaque deux dimensions.

### 2.4. La Matrice de Covariance

Dans le cas où notre ensemble de données contient plus de deux dimensions, plusieurs covariances peuvent être calculées. En général, pour un ensemble de « n » dimensions, on peut calculer :  $\frac{n!}{(n-2)!2}$  covariances différentes. Ces dernières seront mises dans une matrice appelée « Matrice de Covariance » qui est de taille  $n \times n$ . Les caractéristiques de cette matrice sont :

- Les éléments de la diagonale représentent des « Variances »,
- La ligne « i » et la colonne « j » représentent la covariance calculée entre la  $i^{ème}$  et la  $j^{ème}$  dimension,
- C'est une matrice symétrique.

#### **Exemple :**

$$Matrice\ Covariance = \begin{pmatrix} Cov(X, X) & Cov(X, Y) & Cov(X, Z) \\ Cov(Y, X) & Cov(Y, Y) & Cov(Y, Z) \\ Cov(Z, X) & Cov(Z, Y) & Cov(Z, Z) \end{pmatrix}$$

## 3. Présentation du principe général de la PCA

Posons la matrice « X » Suivante de taille « n × p » :

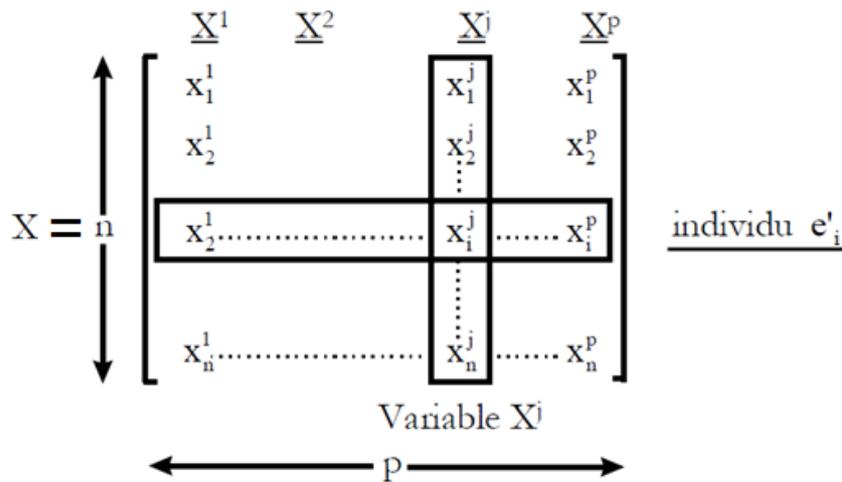


Figure 4-1 : Matrice X de taille  $n \times p$ .

Où :

- « n » : Représente le nombre d'individus, et donc des lignes,
- « p » : Représente le nombre de variables observées pour chaque individu (Pour le cas de la RV, «  $e_i$  » sera un vecteur généré par la concaténation de la matrice de l'image de l'individu « i », on aura donc «  $p = L \times C$  », où L et C sont le nombre de lignes et de colonnes respectivement),
- $\mathcal{R}_p$  et  $\mathcal{R}_n$  représentent l'espace d'individus et de variables respectivement.

### 3.1. Représentation du nuage des individus dans l'espace $\mathcal{R}_p$

A chaque individu noté «  $e_i$  », on peut associer un point dans  $\mathcal{R}_p =$  espace des individus. La Figure 4-2, montre la représentation d'un individu « i » dans un espace  $\mathcal{R}_3$  :

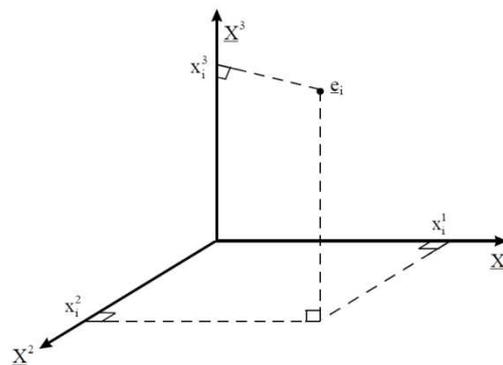


Figure 4-2 : Représentation d'un individu « i » dans  $\mathcal{R}_3$ .

**Remarque :**

- à chaque variable du tableau X est associé un axe de  $\mathcal{R}_p$ .
- Impossible de représenter graphiquement dès que  $P > 3$ , d'où l'utilité de la PCA.

### 3.2. Le Principe général de la PCA

On cherche une représentation des « n » individus  $\{e_1, e_2, e_3 \dots e_n\}$  dans un sous-espace  $\mathcal{F}_K$  de  $\mathcal{R}_p$  de petite dimension « K » (Exemple :  $K=2$ , un plan). Autrement dit, on cherche à définir « K » nouvelles variables combinaisons linéaires des « p » variables initiales qui feront perdre le moins d'informations possibles.

- Ces variables seront appelées « **composantes principales** »,
- les axes qu'elles déterminent : « **axes principaux** »,
- les formes linéaires associées : « **facteurs principaux** ».

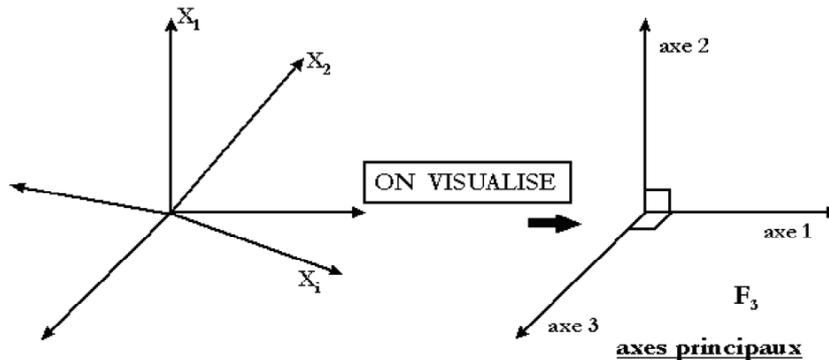


Figure 4-3 : Passage d'un espace  $\mathcal{R}_p$  à un sous-espace  $\mathcal{F}_K$  ( $K \ll p$ ).

Donc  $\mathcal{F}_K$  devra être ajusté le mieux possible au nuage des «  $e_i$  », ce qui implique que **la somme des carrés des distances des «  $e_i$  » à  $\mathcal{F}_K$  doit être minimale.**

Donc :  $\mathcal{F}_K$  est le sous-espace tel que le nuage projeté ait une inertie (dispersion) maximale, l'inertie (Equation suivante) est définie comme la somme des carrés des distances des individus au centre de gravité  $g$ . L'inertie mesure la dispersion totale du nuage de points :

$$I_g = \sum_i^n \frac{1}{n} d^2(e_i, g) \tag{4-6}$$

Où : «  $g$  » est le centre de gravité du nuage des «  $e_i$  », et donc l'image moyenne.

**Justificatif :** Projection orthogonale du nuage sur un sous-espace [36]:

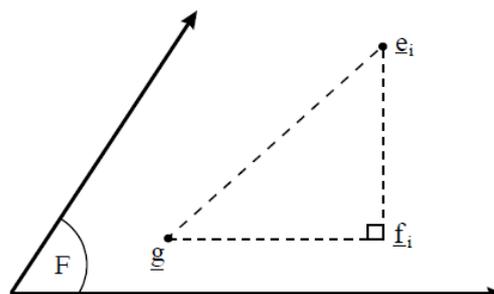


Figure 4-4 : Projection de l'individu «  $e_i$  » sur l'espace  $\mathcal{F}_K$ .

Soit :

- «  $\mathcal{F}$  » un sous-ensemble de  $\mathcal{R}_p$ ,
- «  $f_i$  » la projection orthogonale de «  $e_i$  » sur  $\mathcal{F}$ .

On a le théorème de Pythagore :

$$\|\underline{e}_i - \underline{g}\|^2 = \|\underline{e}_i - \underline{f}_i\|^2 + \|\underline{f}_i - \underline{g}\|^2 ; \quad i = 1, \dots, N \quad (4-7)$$

On cherche «  $\mathcal{F}$  » tel que :

$$\sum_{i=1}^n p_i \|\underline{e}_i - \underline{f}_i\|^2 \text{ soit minimale} \quad (4-8)$$

Ce qui revient d'après le théorème de **Pythagore à Maximiser l'inertie (dispersion) du nuage projeté** :

$$\sum_{i=1}^n p_i \|\underline{f}_i - \underline{g}\|^2 \quad (4-9)$$

Car la quantité suivante est fixe :

$$\sum_{i=1}^n p_i \|\underline{e}_i - \underline{g}\|^2 \quad (4-10)$$

#### 4. Présentation de l'algorithme PCA pour la reconnaissance de visages [37]

L'Algorithme PCA (Analyse en Composantes Principales) peut être utilisé pour des méthodes de reconnaissance globale de visages. Cet Algorithme consiste en une projection des images sur un sous-espace généré par les visages propres appelé « Espace Propre ».

L'algorithme de PCA inclut deux grandes étapes : une phase d'apprentissage et une autre de classification :

##### 4.1. Phase d'apprentissage

Cette première étape est nécessaire au moins une fois lors de la programmation du système biométrique, durant cette phase neuf opérations sont assurées :

##### *Op 1 : Acquisition des images des visages pour la BDD*

Cette première étape consiste en la création de la BDD, pour chaque image acquise, une matrice est générée. Les éléments de cette dernière représenteront les pixels de l'image.

Dans notre PFE, les images sont à niveaux de gris codés sur 8 bits, donc chaque élément de la matrice a une valeur entre 0 et 255 (Figure 4-5).



(a)

74	141	214	237	234	229
234	235	228	222	223	224
229	224	221	224	225	225
219	222	222	221	222	212
220	217	213	210	204	213
221	218	206	200	201	198
181	178	186	178	159	157
122	133	137	129	120	96
34	57	88	74	48	20
19	32	156	152	18	19
35	53	201	241	59	28

(b)

Figure 4-5 : (a) Image d'un visage en niveaux de gris, (b) : Une partie de la matrice générée.

Donc pour « M » Images de visages on aura :

$$I_i = \begin{pmatrix} Y_{11} & \cdots & Y_{1C} \\ \vdots & \ddots & \vdots \\ Y_{L1} & \cdots & Y_{LC} \end{pmatrix}_{L \times C}, \quad \text{For } i = 1, 2 \dots M \quad (4-11)$$

Avec :

- L : Le nombre de lignes de la matrice  $I_i$ ,
- C : Le nombre de colonnes de la matrice  $I_i$ .

**Remarque :** tout les visages doivent être centrés dans les images qui doivent être de la même taille.

### Op 2 : Concaténation des images

Pour faciliter la manipulation de ces images, chaque matrice est concaténée en une colonne de taille :  $(L \times C) \times 1$ , ou une ligne de taille  $1 \times (L \times C)$ ,

$$\Gamma_i = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{L1} \\ Y_{12} \\ \vdots \\ Y_{L2} \\ \vdots \\ Y_{LC} \end{pmatrix}_{(L \times C) \times 1} \quad (4-12)$$

### Op 3 : Calcul de l'image moyenne

L'image moyenne est calculée par l'équation 4-13, Cette dernière peut être vue comme le centre de gravité du jeu d'images [32] :

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (4-13)$$

*Op 4 : Centrage des images de la base de données*

L'image moyenne ne porte pas d'informations, cette étape consiste à soustraire cette dernière de toutes les images de la BDD.

$$\phi_i = (\Gamma_i - \Psi)_{(LxC) \times 1} \quad (4-14)$$

La matrice A contiendra donc tous les vecteurs centrés alignés colonne-colonne.

$$A = [\phi_1 \ \phi_2 \ \phi_3 \ \dots \ \phi_M]_{(LxC) \times M} \quad (4-15)$$

*Op 5 : Calcul de la matrice de covariance*

Pour le calcul de la matrice de covariance (Voir Annexe X : Rappels Mathématiques), l'équation 4-16 est utilisé :

$$C = \frac{1}{M} \sum_{n=1}^M (\phi_n \phi_n^T) = (AA^T)_{(LxC) \times (LxC)} \quad (4-16)$$

*Op 6 : Calcul des valeurs et vecteurs Propres de la matrice covariance*

On voit bien, que la matrice de covariance est de très grande taille :  $(L \times C) \times (L \times C)$ , Le calcul nécessaire pour avoir les  $(L \times C)$  Valeurs et vecteurs propres est très long et prend donc énormément de temps.

Pour contourner cette contrainte, on utilisera la matrice  $A^T A$  de taille  $(M \times M)$  au lieu de la matrice  $AA^T$  ( $M \lll L \times C$ ), le développement suivant est donc suffisant :

On considère :

$V_i, \mu_i$ : Vecteurs et valeurs propres respectivement de la matrice  $A^T A$  :

$$A^T A V_i = \mu_i V_i \quad (4-17)$$

On multiplie des deux termes de l'équation par la matrice A :

$$A A^T A V_i = \mu_i A V_i \quad (4-18)$$

Posons :  $C = A A^T$ , donc :

$$C A V_i = \mu_i A V_i \quad (4-19)$$

Posons :  $U_i = A V_i$ , donc :

$$C U_i = \mu_i U_i \quad (4-20)$$

On remarque bien que les  $U_i$  sont les vecteurs propres de la matrice C et les  $\mu_i$  sont les valeurs propres de la même matrice. On peut donc conclure :

- Les matrices  $AA^T$  et  $A^T A$  ont les même valeurs propres, et leurs vecteurs propres sont reliés par l'équation :  $U_i = A V_i$ .
- Le calcul du vecteurs propres  $V_i$  est plus facile est plus rapide car la matrice  $A^T A$  est de taille inférieure à la matrice  $AA^T$ .

Les vecteurs propres trouvés sont ensuite ordonnés selon leurs valeurs propres correspondantes, de manière décroissante [32].

**Remarque :** Les valeurs propres  $\mu_i$  représentent la variance des visages contenus dans la BDD après leurs projections sur le  $i^{\text{ème}}$  visage propre correspondant à  $\mu_i$ . Les valeurs propres ayant une grande valeur représentent une large variation des caractéristiques des éléments, contrairement aux valeurs propres ayant une petite valeur. On peut réduire donc la taille de l'espace propre, en éliminant :

- Les visages propres possédant des valeurs propres petites, car ils ne présentent pas beaucoup d'informations pour distinguer et différencier les individus, ces valeurs propres sont dues aux bruits.
- Les visages propres ayant des valeurs propres extrêmement élevés, car ces valeurs sont dues à l'éclairage des images qui est très commun.

L'élimination de ces deux cas, n'influera pas trop sur les performances du système, mais permettra de minimiser le temps de la reconnaissance en réduisant la taille de l'espace propre. Plus d'informations seront présentées dans le chapitre 5.

### Op 7 : Calcul des visages Propres

Posons :

$$L = A^T A \quad (4-21)$$

Le vecteur propre  $V_i$  indique la relation linéaire entre les images des visages et le visage propre  $U_i$

$$U_I = \sum_{k=1}^M V_{Ik} \phi_k, \quad \text{pour : } I = 1 \dots M \quad (4-22)$$

### Op 8 : Calcul des poids des visages propres dans l'espace propre

Toutes les images des visages de la base de données sont projetées dans le sous-espace généré par les M visages propres, pour avoir leurs poids:

$$W_{Ik} = U_k^T (\Gamma_I - \Psi), \quad \text{pour : } k, I = 1, 2, \dots M \quad (4-23)$$

$$W_{Ik} = U_k^T \phi_I \quad (4-24)$$

Les « M » poids sont rangés dans une seule matrice comme suit :

$$\Omega_i^T = [W_{i1} \quad W_{i2} \quad \dots \quad W_{iM}], \quad \text{pour : } i = 1, 2, \dots M \quad (4-25)$$

### Op 9 : Calcul du seuil :

Le seuil est une opération additionnelle pour l'amélioration de la performance du système biométrique, il est déterminé expérimentalement pour avoir un FAR = FRR.

## 4.2. Phase de Classification

C'est la phase « Utilisation » du système biométrique, L'algorithme PCA est présenté dans notre PFE pour une méthode d'identification, on mesure donc toutes les distances euclidiennes entre l'image à identifier et celles de la BDD, le minimum de ces distances sera retenu et comparé à un seuil avant de prendre une décision. Dans le cas d'une authentification, une seule distance euclidienne sera calculée et comparée au seuil (Chapitre 1).

Les opérations de la phase de classification sont :

*Op 1 : Soustraction de l'image moyenne*

Comme pour la phase d'apprentissage, l'image est centrée :

$$\phi_{test} = \Gamma_{test} - \Psi \quad (4-26)$$

*Op 2 : Calcul des poids de l'image test dans l'espace propre*

Une fois l'image centrée, elle sera projetée sur l'espace propre, on aura donc « M » Poids pour cette image :

$$W_k = U_k^T (\Gamma_{test} - \Psi), \text{ pour } : k = 1 \dots M \quad (4-27)$$

Les M poids sont stockés colonne-colonne dans la matrice suivante :

$$\Omega_{test}^T = [W_1 \quad W_2 \quad \dots \quad W_M] \quad (4-28)$$

*Op 3 : Calcul de la distance Euclidienne*

Pour pouvoir prendre une décision, les distances Euclidiennes entre les poids de l'image test et ceux de la base de données sont calculées par l'équation suivante :

$$\varepsilon_k^2 = \|\Omega_{test} - \Omega_k\|^2, \text{ pour } : k = 1 \dots M \quad (4-29)$$

Avec:

- $\Omega_k$  : Vecteur qui décrit la  $K^{ème}$  Classe de visage.

La méthode la plus simple pour déterminer quelle classe de visage fournit la meilleure description d'une image d'entrée est de trouver la classe de visages k qui minimise la distance euclidienne.

Un visage appartient à une « classe K » quand le minimum  $\varepsilon_k$  est en dessous d'un certain seuil  $\theta$ . Dans le cas contraire, le visage est classé comme étant inconnu et peut éventuellement être utilisé pour créer une nouvelle classe de visages.

**Exemple :** Prenons un espace propre généré par deux vecteurs propres ( $u_1$  et  $u_2$ ) et donc un « Plan propre » contenant trois classes d'individus connus (1, 2, 3), l'image peut alors :

- **Appartenir au plan propre :** Dans ce cas l'image est une image de visage, deux cas se présentent :
  - L'image appartient à une classe d'individus : l'image test est assignée à l'individu de cette classe (Figure 4 6, Cas1),
  - L'image n'appartient pas à une classe d'individus : l'image représente un individu inconnu dans la BDD (Figure 4 6, Cas 2).
- **Ne pas Appartenir à cette espace :** L'image n'est donc pas une image de visage, une projection de l'image sur le plan propre est nécessaire, deux cas se présentent aussi :
  - La projection de l'image test appartient à une classe d'individus : l'image est éloignée de notre espace, mais sa projection appartient à une classe connue, on parle dans ce cas alors d'une fausse acceptation (Figure 4 6, Cas 3),
  - La projection de l'image test n'appartient pas à une classe d'individus : l'image est éloignée aussi de notre plan propre et sa projection l'est aussi. L'image test ne sera assignée à aucun individu (Figure 4-6, Cas 4).

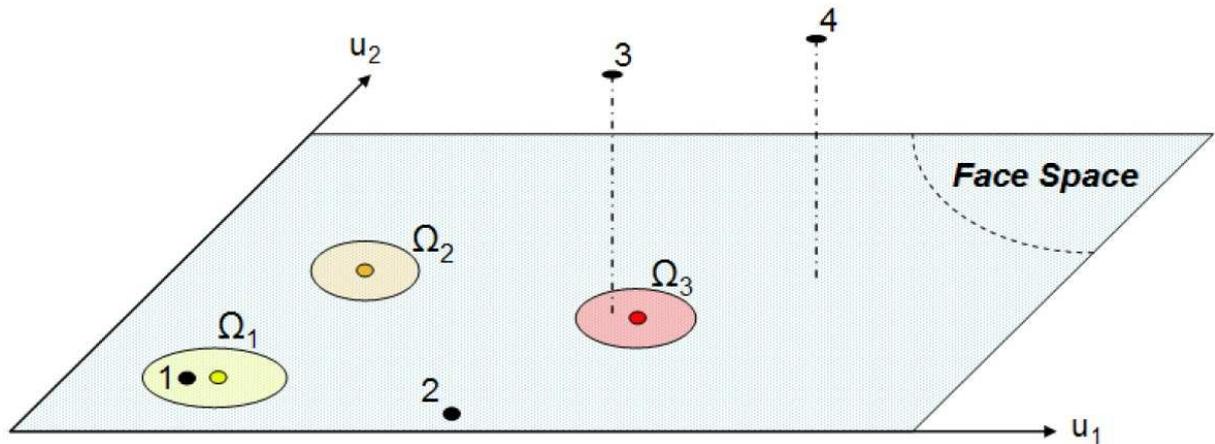


Figure 4-6 : Une version simplifiée de l'Espace propre illustrant les quatre résultats de la projection d'une image sur ce dernier. [32].

#### Op 4 : Etape décision

Une fois les distances Euclidiennes Calculées, l'image testée sera assignée au visage de la BDD ayant la distance Euclidienne minimale si cette dernière est inférieure au seuil.

Donc si :

- $\min\{\varepsilon_k\} > \theta$  ----> Le visage n'existe pas dans la BDD.
- $\min\{\varepsilon_k\} \leq \theta$  ----> L'image sera assignée à l'image de la personne ayant une distance euclidienne minimale avec l'image test.

**Remarque :** Pour notre PFE, la méthode de la distance Euclidienne est utilisée, mais il existe d'autres méthodes pour le calcul de la ressemblance entre deux points, on peut citer :

1. L1 Distance de Manhattan :

$$d(x, y) = |x - y| = \sum_{i=1}^k |x_i - y_i| \quad (4-33)$$

2. L2 Distance euclidienne (au carré) :

$$d(x, y) = \|x - y\| = \sum_{i=1}^k |x_i - y_i|^2 \quad (4-31)$$

3. Angle négatif entre les vecteurs d'image

$$d(x, y) = -\frac{x \cdot y}{\|x\| \|y\|} = -\frac{\sum_{i=1}^k x_i y_i}{\sqrt{\sum_{i=1}^k (x_i)^2 \sum_{i=1}^k (y_i)^2}} \quad (4-32)$$

4. Distance de Mahalanobis (Formule simplifiée):

$$d(x, y) = -\sum_{i=1}^k \frac{1}{\sqrt{\mu_i}} x_i y_i \quad (4-33)$$

Avec  $\mu_i$ , la valeur propre correspondant au  $i^{\text{ème}}$  vecteur propre.

**Remarque :** Pour plus de lectures sur les distances utilisées pour la mesure de similarités, le lecteur pourra consulter la référence. [42]

## 5. Inconvénients de la méthode

---

Aucune méthode n'est parfaite ; malgré ses qualités, la méthode ACP a quelques inconvénients, comme :

- Condition d'éclairage : La variation d'éclairage entre les photos de tests et celles de la BDD causera une diminution importante de performance du système,
- Echelle : les performances décroissent rapidement avec le changement de la taille de la tête dans l'image ... etc. [34]

Une solution peut être développée pour minimiser l'effet de ces inconvénients, c'est la normalisation des données.

## 6. Conclusion

---

L'algorithme PCA utilisé dans notre PFE est une méthode globale utilisant en premier lieu les niveaux de gris des pixels d'une image.

Le principe selon lequel on peut construire un sous-espace vectoriel en ne retenant que les vecteurs propres les plus significatifs, tout en conservant beaucoup d'information utile, fait du PCA un algorithme efficace et couramment utilisé en réduction de dimensionnalité.

Enfin, l'étude théorique de l'algorithme PCA est très pédagogique et permet d'acquérir de solides bases pour la reconnaissance 2D du visage. C'est un algorithme incontournable!

# Chapitre 5 : Simulation en Matlab des deux algorithmes et de leur combinaison.

## 1. Introduction

Ce chapitre est dédié à la partie programmation des deux algorithmes PCA et WPD sur MATLAB de notre PFE. Le recours à ce langage, est essentiellement pour la validation et la vérification des deux algorithmes déjà vus dans les chapitres 3 et 4.

On commence le chapitre par une présentation de la base de données utilisée. Ensuite, la description des étapes suivies pour la programmation des algorithmes PCA et WPD et leur combinaison en donnant aussi quelques résultats obtenus lors de la simulation.

On terminera le chapitre par l'étude des deux solutions proposées pour améliorer les taux du système RV (taux d'identification, FAR, FRR).

## 2. Présentation de la Base de données (BDD)

La base de données utilisée est créé à l'aide d'une caméra fixe. Plusieurs photos sont prises pour chaque personne pour avoir le maximum de variations d'expressions faciales possibles de cette personne.

Cette BDD a été proposée par « Dr Libor Spacek » qui est maître de conférences en vision par ordinateur dans le département des sciences informatique à l'Université d'Essex en Angleterre, la BDD peut être téléchargée du site web suivant : <http://cswww.essex.ac.uk/mv/allfaces>.

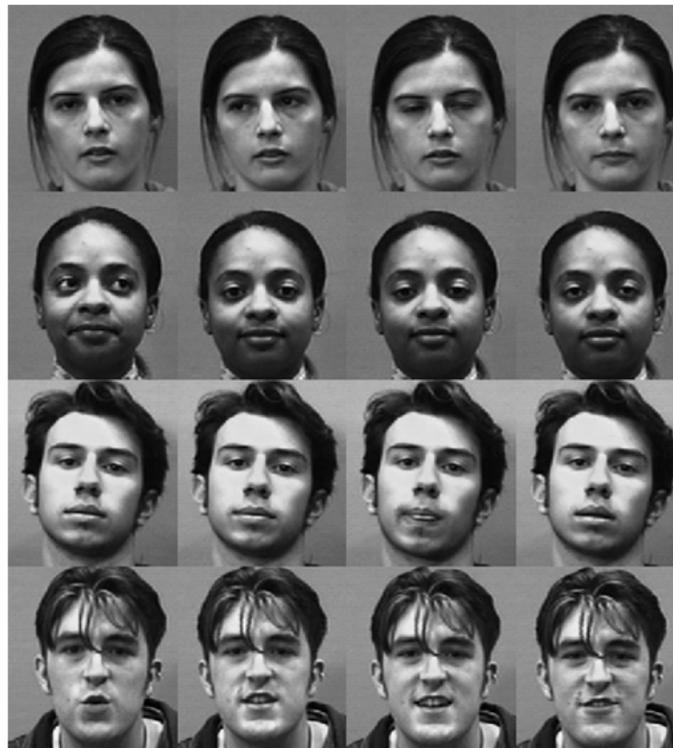


Figure 5-1 : Quelques visages de la BDD utilisée.

## 2.1. Descriptions de la BDD

- Nombre d'individus : 132.
- La résolution d'image : 200x180 Pixels.
- Genre : 20 femmes et 112 hommes.

## 2.2. Variations présentes dans les images d'individus

- Arrière-plan: Fond vert uni.
- Echelle de la tête : Invariable.
- Rotation de la tête, l'inclinaison et l'angle : très faibles variations de ces paramètres.
- Position du visage dans l'image : Généralement centré, avec de petites variations.
- La variation d'éclairage d'image : Aucune.
- Variation d'expressions : Considérable.

**Remarque :** Les images de cette base de données ont été prises en une seule séance.

## 3. L'Analyse en Composantes Principales (ACP) sur Matlab

La programmation de cet algorithme est faite d'une façon modulaire sur plusieurs scripts Matlab, cela facilitera la modification des différents paramètres utilisés dans l'ACP (le nombre d'éléments de la BDD, le nombre de visages propres ...).

Avant d'entamer l'ACP, Le script « *Lect\_base.m* » fera la lecture des images de la BDD, qui seront enregistrées sous forme de vecteurs dans le *work* de Matlab.

Le Script « *Lect\_base.m* » :

```

% ----- Lecture de la base -----
%{
  But :
  - Lecture des images de la base de données
  - Mettre les différentes images dans une seule matrice A
%}
timeZ=[];
% lecture d'une image pour calculer le size
img=rgb2gray(imread('..\image\database1\Person1\1.jpg'));
[l,c]=size(img); %la taille d'une SEULE image
for Z=1:3 % Nombre d'image par personne (doit etre Z<20 dans notre cas)
  tic;
  clear VA; % effacer seulement VA
  clc;
  M=132; % Nombre d'individu pris de la BDD
  % Lecteur des images de la base
  VA=[];
  for i=1:M
    for j=1:Z
      str=strcat('..\image\database1\Person',int2str(i),'\',int2str(j),'.jpg')
      eval('img=im2single(rgb2gray(imread(str)));');
      %eval('img=im2double(rgb2gray(imread(str)));'); %variables doubles
      V=Qanka(img,1,1,c); % Appel de Qanka.m (resultat : vecteur colone)
      VA=[VA V];
    end
  end
  t=toc;
  timeZ=[timeZ t];
  % Sauvgarde des résultats dans data_zX
  eval(strcat('save data1_z',int2str(Z),' VA 1 c M Z timeZ'));
end

```

Figure 5-2 : Image du scripte Matlab Lect\_base.m.

### 3.1. Etape n°1 : Concaténation des images

L'ACP exige que les matrices soient concaténées en un vecteur ligne ou colonne. Pour cela on a créé une fonction appelée « **Qanka.m** ».

La sortie de cette fonction dépend du second paramètre « *i* » de cette fonction tel que :

- Si « *i* = 1 » la sortie est un vecteur de taille  $L \times C$  ce qui correspond au nombre de pixels de l'image,
- Si « *i* = 2 » la sortie dans ce cas est une matrice reconstruite après la déconcaténation du vecteur présent à l'entrée.

Après concaténation de toutes les images de la BDD l'ensemble d'images de visages de la BDD est maintenant considéré comme un ensemble de points dans un espace de dimension  $L \times C$ .

Le script « **Qanka.m** » :

```

%% Cette fonction assure la quancatination et son operation inverse

function [ y ] = Qanka( x, i,l,c )
% i = 1 : Operation quancatination : y = vecteur et x = matrice
% i = 2 : Operation inverse : y = matrice et x = vecteur
%% ----- Quancatination -----
if i==1
    y=[];
    for j=1:c
        y=[y ; x(:,j)];
    end
end
%% ----- Dequancatination -----
if i==2
    y=[];
    for i=1:c
        y=[y x(1+(i-1)*1:i*1,1)];
    end
end
end
end
    
```

Figure 5-3 : Image du script Matlab de la fonction Qanka.m.

### 3.2. Etape n°2 : Calcul des visages propres

Après avoir concaténé les différentes images des visages de la BDD, l'ensemble des vecteurs est mis en une seule matrice pour calculer les visages propres de cette dernière comme suit :

- Calcul de l'image moyenne de la BDD (Figure 5.5) : Soit  $N$  le nombre de visages de la BDD et  $\Gamma_i$  le vecteur représentant le visage d'indice « *i* », l'image du visage moyen est :

$$\phi = \frac{1}{N} \sum_{j=1}^N \Gamma_j \tag{5-1}$$

- Soustraction de l'image moyenne des images de la BDD, l'équation suivante est utilisée :

$$\hat{\Gamma}_j = \Gamma_j - \phi. \tag{5-2}$$

Les résultats sont stockés dans la matrice  $A$  comme suit :

$$A = [\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_N] \quad (5-3)$$

- Calcul de la matrice de covariance  $COV$  et des valeurs et vecteurs propres de  $COV$ : On doit calculer la matrice  $COV = A^T A$  de taille  $N \times N$  pour pouvoir calculer les valeurs et les vecteurs propres (Voir Chapitre 4).

On obtient donc  $N$  valeurs propres  $\lambda_i$  et  $N$  vecteurs propres  $U_i$  de dimension  $N \times 1$ .

- Calcul des visages propres du système : Afin d'obtenir les visages propres du système utilisé pour la génération de l'espace propre, il suffit de multiplier les vecteurs propres  $U_i$  par la matrice  $A$

$$VP_i = A \times U_i \quad (5-4)$$

**Exemple :** On prend une BDD de  $N = 132$  personnes, chaque personne est représentée par une seule image ( $Z = 1$ ).

- L'image moyenne :



Figure 5-4 : image moyenne de la BDD contenant 132 personnes

- Valeurs propres :

Les résultats du calcul des valeurs propres sont présentés par ordre décroissant sur Matlab. Les dernières valeurs propres étant moins significatives et ne donnant pas beaucoup d'informations peuvent être ignorées (Figure 5-5)

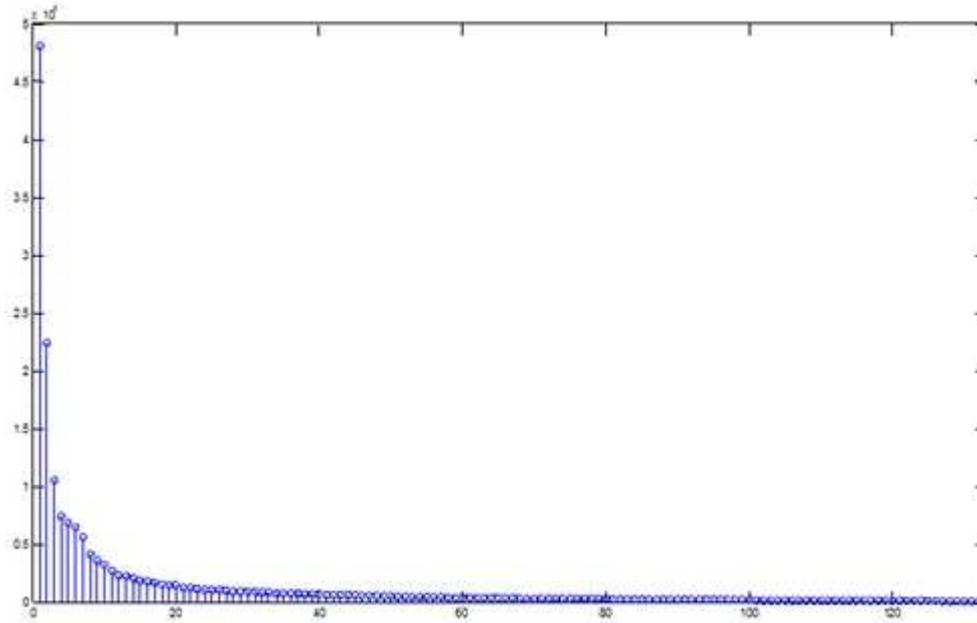


Figure 5-5 : Variation des valeurs propres de notre BDD.

D'après la Figure 5-5, on peut constater que seulement les 10 à 15% premières valeurs propres ont une signification importante et peuvent être utilisées pour représenter nos données avec le moins d'éléments possibles et faire une bonne distinction entre ces derniers. Donc seules ces premières valeurs seront utilisées pour la génération de notre espace propre,

➤ Visages propres :

La BDD contient  $N=132$  éléments, donc il y a 132 visages propres ; leur importance n'est pas identique car ils ne portent pas la même quantité d'informations. En effet, seules les premiers visages propres correspondant aux valeurs propres  $\lambda_i$  les plus grandes seront gardés pour constituer l'espace propre. La Figure 5-6 montre les 12 premiers visages propres de la BDD.



Figure 5-6 : Les 12 premiers visages propres de notre BDD.

### 3.3. Etape n°3 : Génération de l'espace propre

Cette étape est très importante car elle influera sur la performance du système côté précision des résultats et aussi sur le temps mis pour prendre une décision (temps de traitement).

Dans cette étape on choisit le nombre de visages propres à garder pour réaliser notre système de RV et générer l'espace propre. Après le choix du nombre de visages propres une projection

de la BDD sur l'espace propre est effectuée et les poids obtenus sont enregistrés pour pouvoir faire des tests de comparaison par la suite avec les images tests.

Pour accroître la performance du système pour la détection des imposteurs, On définit :

- Un seuil  $\theta$  qui nous permettra d'avoir des taux de « Fausse Acceptation » et « Faux Rejet » faibles. La valeur de ce seuil dépend de plusieurs paramètres comme le nombre de visages propres utilisés, le nombre de personnes présentes dans la BDD ... La valeur de ce seuil est déterminée expérimentalement et est choisie en général pour une valeur qui donne un  $FAR \cong FRR$ , ceci n'est pas une règle, il existe des applications qui exigent un niveau de sécurité très élevé, on préfère dans ce cas rejeter des utilisateurs légitimes que d'accepter des imposteurs donc avoir un  $FAR < FRR$ .
- Un intervalle de confiance « INTC » qui délimite l'espace des visages propres par deux paramètres qui sont : le module max «  $P_{max}$  » et min «  $P_{min}$  » des poids des visages de la BDD. L'intervalle INTC sera défini comme suit :

$$INTC = [P_{min}, P_{max}] \quad (5-5)$$

Cet intervalle nous permettra principalement de juger si une image représente un visage ou non, et si cette personne est présente dans la BDD ou non sans avoir à calculer plusieurs distances euclidiennes.

Les deux limites supérieures et inférieures de l'intervalle « INTC » peuvent avoir en plus et en moins respectivement un certain seuil  $\rho > 0$ .

Le module de toute image test projetée dans l'espace propre est calculé et s'il n'appartient pas à l'intervalle « INTC », l'image sera rejetée directement et la procédure de reconnaissance est interrompue.

### 3.4. Etape n°4 : Phase de classification

Pour vérifier le bon fonctionnement de notre système biométrique basé sur PCA, plusieurs tests sont effectués, en utilisant des images de visages de personnes appartenant à la BDD et d'autre non. La phase de classification comporte les étapes suivantes :

- Concaténation ensuite centrage de l'image à tester,
- Projection de l'image sur l'espace propre pour trouver les différents poids de l'image. La projection sur un visage propre  $VP_i$  donne le poids  $w_i$  qui est défini comme suit :

$$w_i = VP_i^t (\Gamma_{test} - \phi) \quad (5-6)$$

Si le nombre des Visages Propres gardés est «  $P$  », on aura donc «  $P$  » poids à calculer pour chaque image de test, ces poids représentent les coordonnées de l'image test dans l'espace propre.

Pour pouvoir prendre une décision sur une image, deux calculs se font :

- Le module de l'image test qui doit appartenir à l'intervalle « INTC »,
- Dans le cas où la première condition est satisfaite, les distances euclidiennes entre l'image test et les éléments de la BDD sont calculées. Une fois le minimum de ces distances euclidiennes est connu, il sera comparé au seuil  $\theta$ , deux cas se présentent :
  - $\min\{\varepsilon_k\} > \theta \rightarrow$  Image test rejetée, la personne n'est pas dans la BDD, il faut noter que dans ce cas, on peut avoir un « Faux Rejet » comme erreur,

- $\min\{\varepsilon_k\} \leq \theta \rightarrow$  La personne est reconnue, la personne de la BDD ayant une distance euclidienne minimale est assignée à la personne voulant s'identifier, il faut noter que dans ce cas deux erreurs peuvent se présenter : Fausse Acceptation (Acceptation d'un imposteur) ou une fausse correspondance.

#### 4. Mesure des performances du système RV

Pour évaluer la performance du système biométrique utilisant PCA comme méthode de reconnaissance, plusieurs tests sont nécessaires. Les images de la BDD non utilisées dans la phase d'apprentissage seront mises à l'entrée du système pour l'évaluer en calculant les paramètres suivants :

- FRR : False Rejection Rate, qui se calcule en utilisant des images de personnes présentes dans la BDD du système RV,
- FAR : False Acceptance Rate, qui se calcule en utilisant des images de personnes non présentes dans la BDD du système RV,
- Le taux de bonne correspondance (taux d'identification), qui se calcule en utilisant des images de personnes présentes dans la BDD du système RV.

Pour les deux premiers taux on n'utilisera pas la totalité de la BDD (132 personnes) pour la phase d'apprentissage, car on doit faire plusieurs tests (sur des personnes présentes dans la BDD du système RV et d'autres non) et voir la variation du FAR et du FRR en fonction du seuil  $\theta$ .

Par contre pour le taux d'identification, toute la BDD sera utilisée pour la phase d'apprentissage donc  $N = 132$ .

##### 4.1. Mesure du FRR et du FAR

Dans cette partie on a utilisé une BDD pour l'apprentissage contenant 40 personnes où chacune d'elles possède une seule image, en suivant les étapes citées dans la partie PCA allant de la lecture de la BDD jusqu'à l'étape du choix du nombre de visages propres à garder. Comme déjà mentionné, seules les 10% premières valeurs propres sont les plus importantes, c'est pour cette raison qu'on a choisi  $P = 4$  (nombre de visages propres).

Après avoir choisi le nombre de visages propres à garder, un autre paramètre très important, qui est le seuil, doit être déterminé dans le but d'obtenir un FAR presque égal au FRR. Pour cela plusieurs essais sont effectués sur des images de personnes :

- Appartenant à la BDD (Pour le calcul du FRR).
- N'appartenant pas à la BDD (Pour le calcul du FAR).

Le tableau suivant résume les différents résultats obtenus lors de la recherche du seuil idéal donnant un  $FRR \cong FAR$ . Les deux ensembles de tests contiennent 400 images chacun (40 personnes, chacune possédant 10 images).

<b>Seuil</b>	20.00	10.00	9.50	9.00	8.50	8.00	7.50	7.75	7.65	7.60	7.58	7.59	7.00
<b>FRR</b>	0.00	12.75	16.50	20.50	23.75	25.75	28.25	27.00	27.75	27.75	28.00	28.00	30.50
<b>FAR</b>	100	44.00	41.50	37.75	34.00	31.50	27.25	29.25	28.25	28.00	27.75	28.00	18.04

Tableau 5-1 : Mesure du FAR et du FRR pour une BDD de 40 personnes avec  $P=4$ .

Le Tableau 5-1 montre que le choix du seuil n'est pas une chose très facile, car si le seuil est très faible, le FAR sera faible et donc le FRR sera grand donc on rejettera plusieurs utilisateurs légitimes ; même si on choisit un seuil élevé on aura un autre problème qui est l'acceptation de plusieurs imposteurs, la solution idéale donc (Chapitre 1) est d'essayer d'avoir un seuil qui donne un  $FAR \cong FRR$ . Après plusieurs essais, le tableau 5-1 indique **un seuil égal à 7,59**.

Les deux courbes suivantes montrent la variation du FAR et du FRR en fonction du seuil, et la courbe ROC.

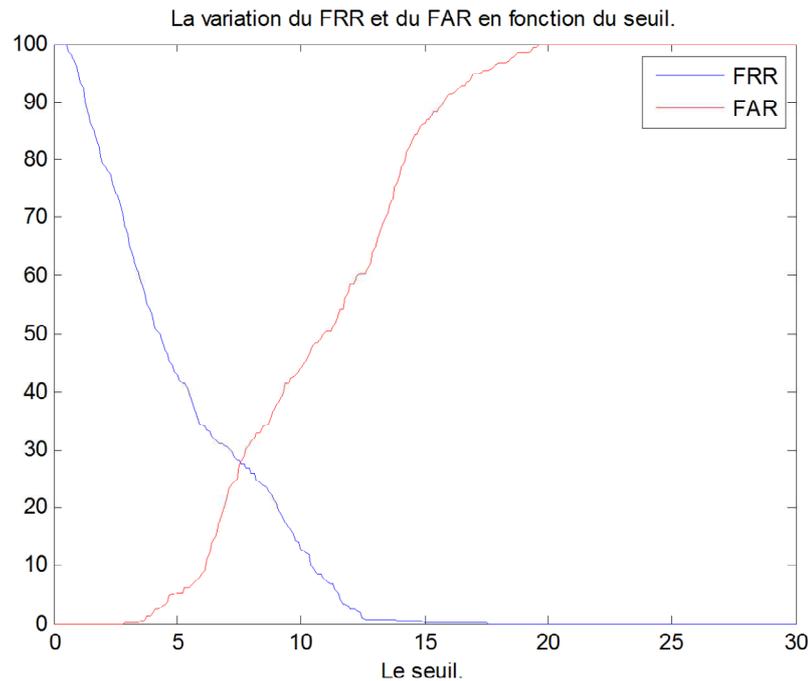


Figure 5-7 : La variation du FAR et du FRR en fonction du seuil.

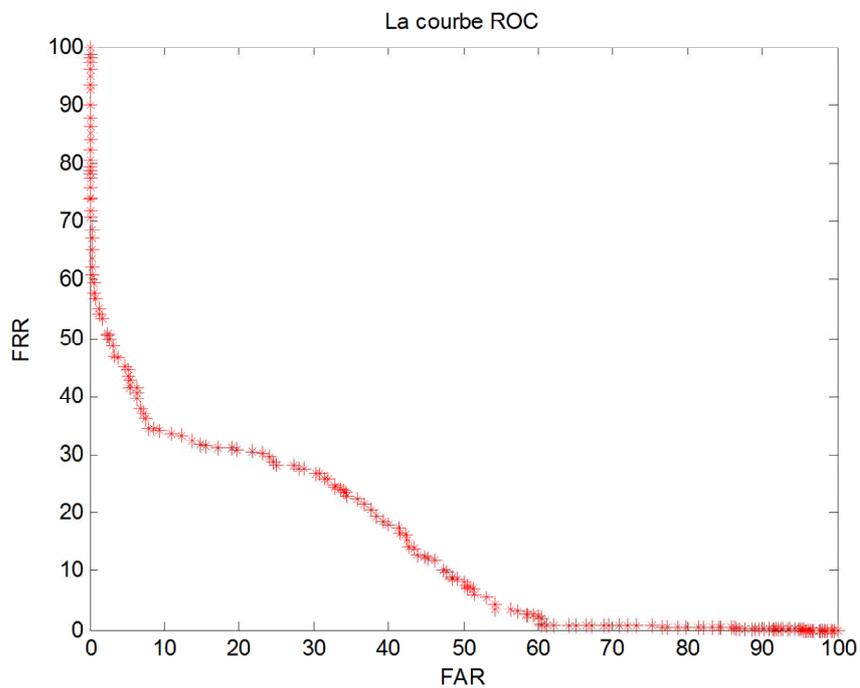


Figure 5-8 : La courbe ROC pour  $P=4$ .

Une fois le seuil déterminé, un autre test est fait pour l'évaluation de ce seuil, ce test peut être aussi réalisé avec deux autres ensembles contenant 360 images chacun (40 personnes, chacune possédant 9 images) pour voir les nouvelles valeurs du FAR et du FRR. Les résultats obtenus sont :

$$FRR = 11.11 ; FAR = 25.83.$$

Si on compare ces deux valeurs aux valeurs du Tableau 5-1 pour le seuil  $\theta = 7.59$ , on constate que le FRR s'est amélioré un peu, chose attendue car les images utilisées pour effectuer son calcul sont des images prises dans un laps de temps petit donc il n'y a pas un grand changement entre ces images et les images utilisées dans la phase d'apprentissage. Par contre le FAR est resté presque égal à la première valeur avec une légère différence.

La courbe ROC obtenue après la mesure du FAR et du FRR est loin d'épouser la courbe idéale, plusieurs solutions peuvent être envisagées pour l'améliorer comme :

- L'augmentation du nombre de visages propres utilisés  $P$ ,
- L'augmentation de la classe de chaque personne (augmenter la classe veut dire augmenter le nombre de visages par personne utilisée dans la phase d'apprentissage  $Z$ ),
- Utiliser la WPD combinée avec PCA (pour éliminer les détails inutiles),
- Combiner les trois méthodes citées ci-dessus.

Les résultats des trois dernières solutions seront vus par la suite après la présentation du travail de la WPD faite sous Matlab.

Pour la première solution qui consiste à augmenter le nombre de visages propres on utilisera  $P = 10$  et avec les mêmes images de la BDD d'apprentissage et images de tests utilisées pour  $P = 4$  on obtient les résultats suivants :

- Les FAR(s) et FRR(s) suivantes :

Seuil	30	20	15	10	12.50	13.50	14.50
FRR	0	1.50	11.75	33	19.50	15.25	12.75
FAR	100	61.75	15.75	2	7.75	8.75	12.75

Tableau 5-2 : Mesure du FAR et FRR pour une BDD de 40 personnes avec  $P=10$ .

- FAR et FRR avec un autre ensemble de test où  $\theta_0 = 14.50$   
 $FRR = 3.05$  ;  $FAR = 20.83$ .
- Les courbes :

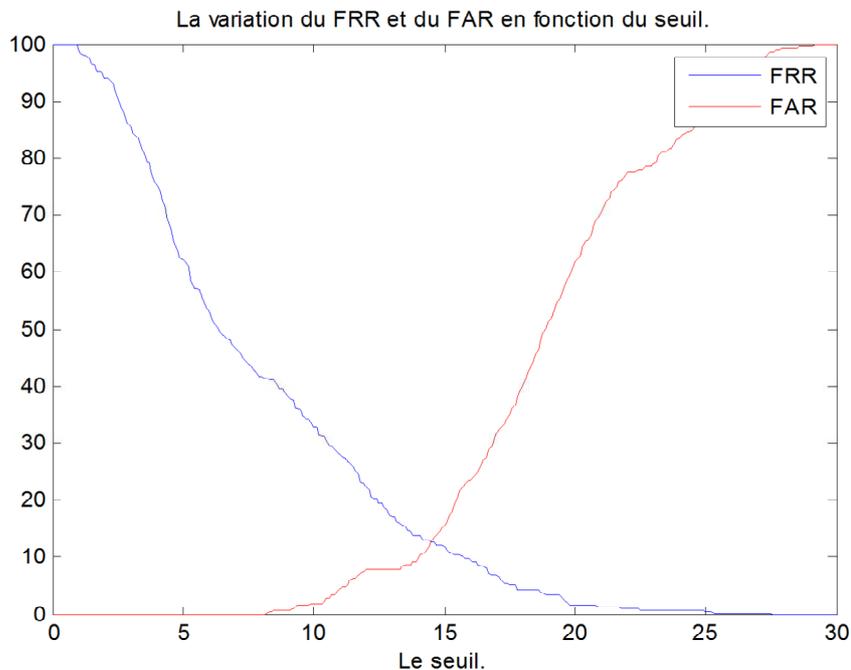


Figure 5-9 : La variation du FAR et du FRR en fonction du seuil

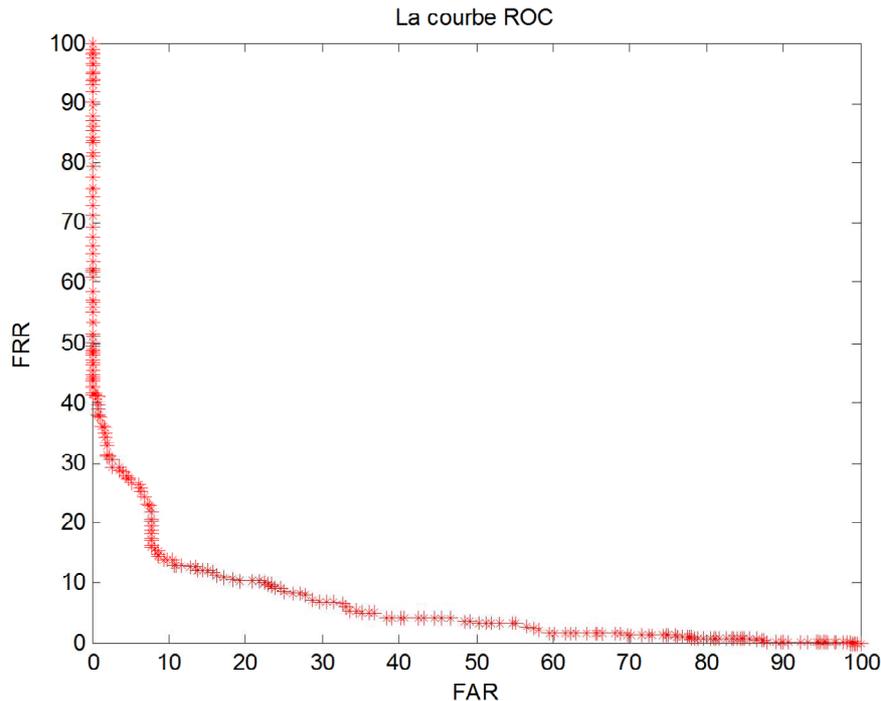


Figure 5-10 : La courbe ROC pour  $P=10$ .

Avec l'augmentation du nombre de visages propres  $P$  la courbe ROC s'améliore et tend à épouser la forme du repère.

D'après les résultats obtenus lors de l'utilisation de  $P = 4$  et  $P = 10$  (nombre de visages propres) on remarque bien une amélioration dans les taux FRR et FAR après avoir augmenté le nombre de visages propres et cela est évident car lorsqu'on augmente le nombre de VP(s) on ajoute de l'information qui aide à mieux faire la reconnaissance mais cela avec l'augmentation du temps de traitement qui sera plus important. Il faut noter que la valeur seuil  $\theta$  déterminée expérimentalement change si un des paramètres principaux de la PCA change, tel que le nombre de personnes de la BDD  $N$  ou bien le nombre de visages propres gardés  $P$  comme montré par les deux essais précédents.

#### 4.2. Mesure du taux d'identification

Le taux d'identification nous indique le nombre de fois où le système a bien reconnu la personne de test. Pour le mesurer on utilise la totalité de la BDD donc le nombre d'images utilisées dans la phase d'apprentissage est égal à 132 (une image pour chaque personne).

Le Tableau 5-3 montre la variation du taux d'identification du système en fonction du nombre «  $P$  » de visages propres utilisés, les images de tests sont les 15 images restantes non prises dans la phase d'apprentissage de chaque personne.

Nombre de visages propres	32	16	14	12	10	8	6	4	2
Taux d'identification	91.06	89.39	90.20	88.28	87.57	85.20	80.00	75.70	55.60

Tableau 5-3 : Taux d'identification en fonction du nombre de visages propres gardés.

La figure suivante montre la variation du taux d'identification en fonction du nombre «  $P$  » de visages propres gardés :

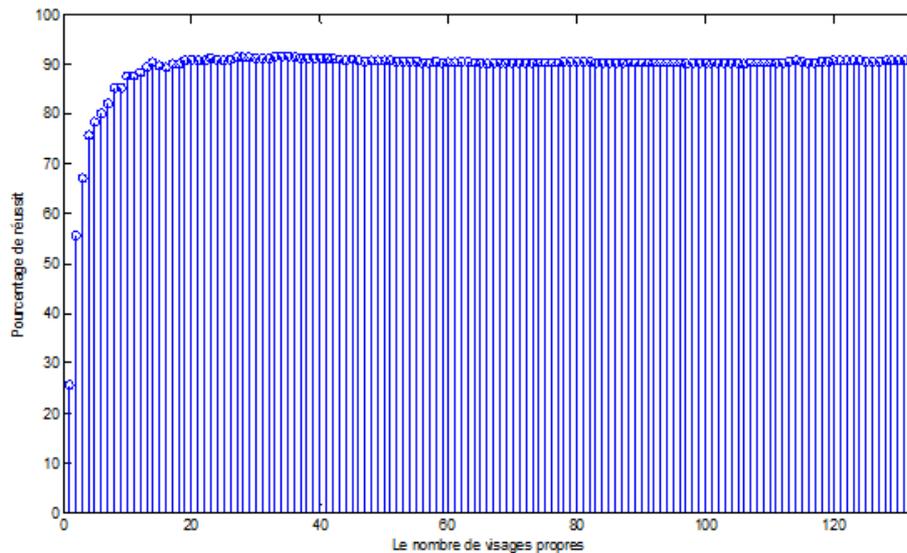


Figure 5-11 : Taux d'identification en fonction du nombre de visages gardés, en utilisant 132x15 images de tests.

On remarque clairement de la Figure 5-5Figure 5-11qu'à partir d'environ 20 visages propres, l'ajout des visages propres n'influe presque pas sur le taux d'identification du système biométrique. De même si on ne prend pas suffisamment de visages propres, la performance du système diminue.

L'autre paramètre très important aussi est le temps nécessaire pour une reconnaissance de visage qui augmente avec le nombre de visages propres gardés.

L'utilisation de la PCA comme algorithme de reconnaissance donne un bon taux d'identification, mais nécessite un temps de traitement important si les images sont de très grande taille ou si la BDD contient un nombre important d'individus. Deux solutions se présentent pour réduire ce temps :

- Réduction du nombre de visages propres gardés : cela causera une diminution de la performance du taux d'identification.
- Utilisation d'un algorithme de compression de données : Cette solution paraît plus intéressante car elle ne causera pas une diminution de performance, mais nécessitera un autre système et donc plus de parties matérielles et logiciels.

## 5. Décomposition d'image en paquets d'ondelettes (WPD)

En utilisant les ondelettes de « Haar » la décomposition en paquets d'ondelettes d'une image revient à calculer la moyenne et la différence des colonnes, ensuite des lignes, deux à deux (chapitre 3), mais comme on s'intéresse seulement à l'image approchée (de résolution plus basse) après décomposition il suffit de calculer la moyenne des lignes deux à deux et ensuite des colonnes deux à deux aussi pour obtenir cette image.

Pour faire cela, on a créé une fonction sur « Matlab » appelé **WPDN** qui reçoit comme paramètres deux éléments :

- A : L'image à compresser,
- n : le niveau de compression désiré.

La sortie de cette fonction est l'image compressée B au niveau indiqué par le paramètre « n ».

La seule contrainte de cette méthode de décomposition est que le nombre d'échantillons constituant notre signal analysé doit être pair (pour l'image, le nombre de lignes et de colonnes doit être pair). La parité ne pose pas de problème à la fonction **WPDN** car cette dernière vérifie le nombre

de lignes et de colonnes de l'image « A » avant de faire la compression. Dans le cas où un des deux nombres est impair, la fonction ajoutera des lignes ou des colonnes à l'image ; ces lignes ou colonnes ajoutées correspondent à des copies de la dernière ligne (en bas) et dernière colonne (à droite) de l'image.

**Exemple :** Ci-dessous l'application de la WPD sur une image avec un niveau de compression « n » égal à 1 :

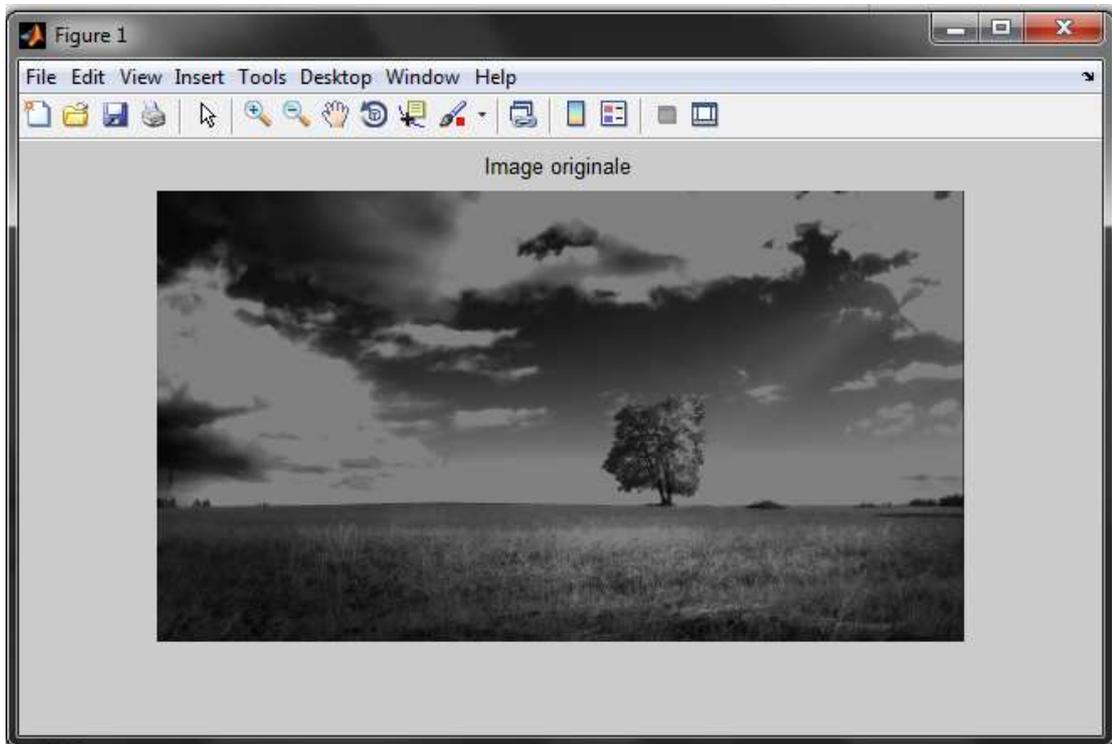


Figure 5-12 : Image originale



Figure 5-13 : Image compressée avec WPD pour n=1.

Pour les images de notre BDD, le niveau de décomposition « n » ne doit pas dépasser la valeur « 7 » car c'est la valeur maximale possible avec les tailles d'images 200 × 180 pixels.

Il faut noter qu'à un certain niveau de décomposition en ondelettes le nombre de pixels représentant une image devient très réduit ce qui influera sur la qualité visuelle de l'image, on passe d'une image claire (l'image originale) à une image de plus en plus floue car à chaque fois on élimine des détails de cette image, jusqu'à atteindre un niveau où l'œil humain ne peut reconnaître la personne.

La Figure 5-14 illustre les résultats obtenus par décomposition d'une image de la BDD de résolution  $200 \times 180$  pixels, pour «  $n$  » variant entre 1 et 7, la Figure 5-15 représente les différentes images qui sont agrandies pour mieux voir leur contenu.



Figure 5-14 : La décomposition d'une image de la BDD dans les sept niveaux possibles.  
Extrême gauche, image originale.

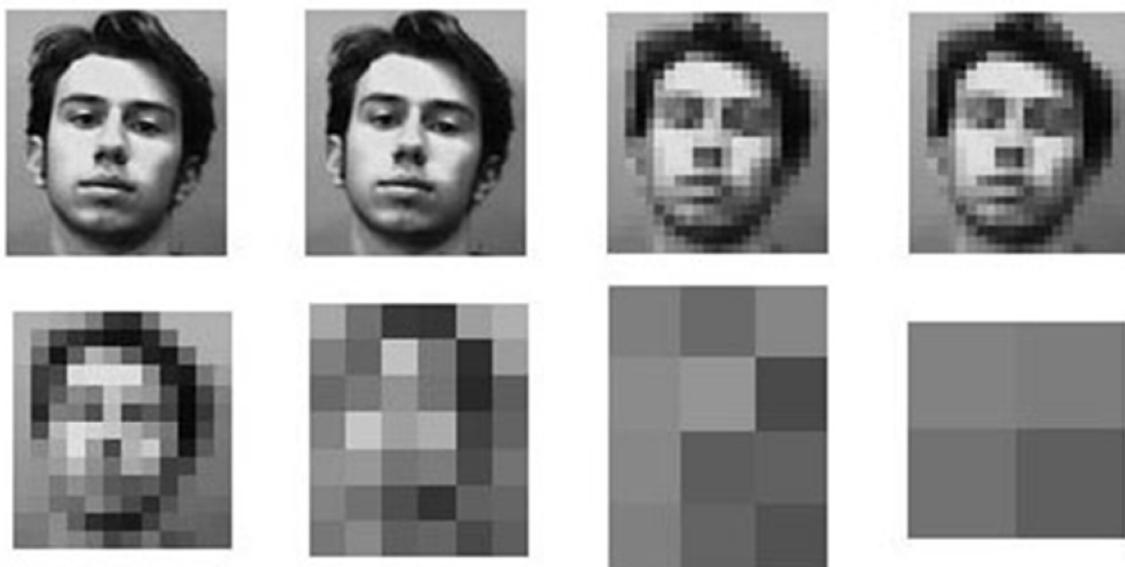


Figure 5-15 : Les images compressées vues avec un zoom.

## 6. La combinaison séquentielle de WPD et PCA

L'avantage de l'utilisation de la compression d'images avec WPD juste avant le système de reconnaissance des visages basé sur PCA est :

- La réduction du nombre de pixels traités,
- L'élimination des détails inutiles pour la phase de reconnaissance et donc une amélioration des différents taux de reconnaissances,
- La réduction du temps de traitement d'une image.

La programmation de cette partie est réalisée par un simple appel à la fonction **WPDN** avant l'exécution des étapes du PCA.

On présentera par la suite quelques résultats obtenus lors de la mesure du FAR, FRR et du taux d'identification en faisant varier le niveau de compression « n » de 1 jusqu'à la valeur 5 car à partir d'un niveau supérieur à 5 on atteint une résolution où même le visage flou disparaît et on obtient seulement des carrés.

### 6.1. Mesure du FRR et du FAR

Utilisant le même principe décrit dans la partie 4.1 de ce chapitre on obtient les résultats suivants pour  $P = 10$  (nombre de VP) et pour  $n$  variant de 1 jusqu'à 5.

- Les seuils  $\theta_i$  qui donnent un  $FAR \cong FRR$  :

Le niveau de compression $n$	0	1	2	3	4	5
Seuil $\theta$	14.50	7.22	3.59	1.76	0.84	0.37
$FRR \cong FAR$	12.75	12.62	12.50	12.12	8.37	6.25

Tableau 5-4 : Mesure du seuil qui donne  $FAR \cong FRR$  en fonction de « n ».

On remarque que lorsqu'on augmente le niveau de compression des images, les taux (FAR et FRR) s'améliorent jusqu'à 50% si on compare le résultat obtenu pour  $n = 5$  et  $n = 0$ . Pour mieux visualiser cette performance la Figure 5-5 illustre les courbes ROC(s) obtenues pour les 5 niveaux de compression et aussi la courbe ROC du système sans compression.

- Les courbes ROC (s) :

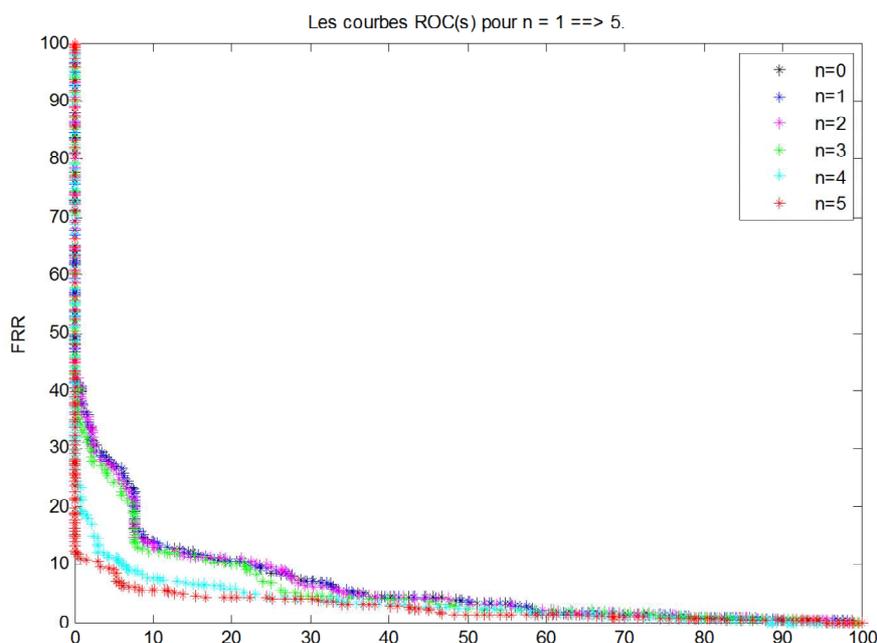


Figure 5-16 : Courbes ROC(s) pour plusieurs niveaux de compression.

On remarque que plus «  $n$  » (niveau de compression) est élevé plus la courbe épouse mieux le repère ROC, et donc plus les performances du système augmente car l'utilisation de la compression par WPD aide à éliminer des détails inutiles qui induisent en erreur le système.

### 6.2. Mesure du taux d'identification

On utilise 14 visages propres car c'est la plus petite valeur (Tableau 5-3) qui donne une bonne performance, le tableau suivant résume les résultats obtenus (avec les mêmes images tests utilisées dans le système PCA) :

Le niveau de compression $n$	0	1	2	3	4	5
Taux d'identification	90.20	90.30	90.70	91.71	93.43	95.85

Tableau 5-5 : Le taux d'identification en fonction du niveau de compression

On remarque du Tableau 5-5 que le taux d'identification s'est amélioré et a atteint un maximum pour  $n = 5$ , cela revient en l'élimination des détails inutiles dans la phase de reconnaissance qui poussent notre système à faire des erreurs de confusion. Avec ce résultat on peut dire que le système RV résultant de la combinaison des deux algorithmes PCA et WPD et se comportant comme un système visuel humain pouvant reconnaître des visages familiers à très basse résolution.

## 7. Amélioration du taux de reconnaissance du système RV

Pour améliorer les différents taux de reconnaissance (FAR, FRR) et d'identification du système, deux solutions sont proposées : la première concerne la partie Apprentissage du système, et la seconde la partie Classification.

### 7.1. Augmentation de classes d'individus

Cela signifie que le nombre d'images par individu «  $Z$  » utilisées dans la phase d'apprentissage sera augmenté pour ajouter une variabilité dans les images du même individu et donc de l'information en plus, ce qui augmentera les différents taux.

#### 7.1.1. Mesure du FRR et du FAR

Utilisant le même principe décrit dans la partie 4.1 de ce chapitre, on obtient les résultats suivants pour  $P = 10$  (nombre de VP) et pour  $Z$  variant de 1 jusqu'à 5.

- Les seuils  $\theta_i$  donnant un  $FAR \cong FRR$  :

Images/individu $Z$	1	2	3	4	5
Seuil $\theta$	14.50	13.81	13.60	13.30	13.10
$FRR \cong FAR$	12.75	10.25	9.87	10.37	10.75

Tableau 5-6 : Mesure du seuil qui donne  $FAR \cong FRR$  en fonction de «  $Z$  ».

- Les courbes ROC (s) :

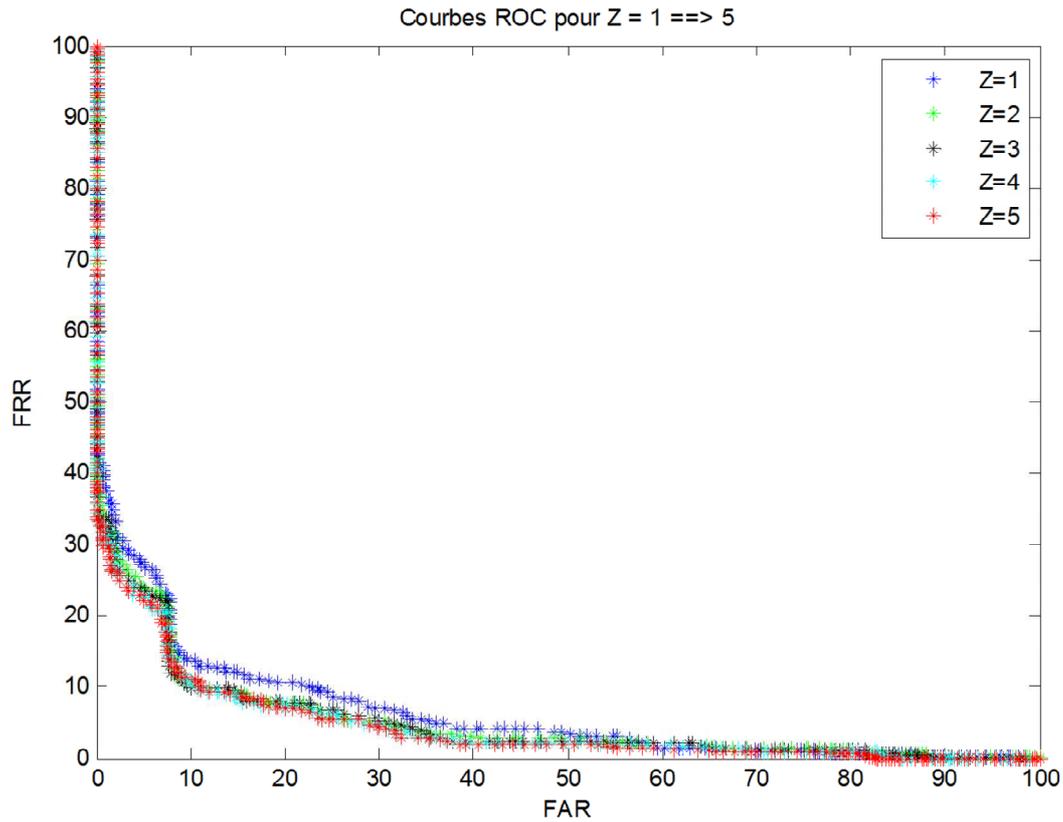


Figure 5-17 : Les différentes courbes ROC(s) obtenues par variation du Z.

On remarque de la Figure 5-17 qu'il y a bien une amélioration lorsque « Z » augmente mais qui est un peu faible si on la compare avec l'amélioration obtenue lorsque « n » le niveau de compression augmente.

On peut aussi envisager une combinaison entre cette technique et la méthode de compression d'images, les résultats suivants sont obtenus pour  $n = 5$  et  $Z = 5$  :

$$\theta = 0.34, FAR = 8.25, FRR = 8.50$$

La Figure 5-18 illustre les courbes ROC(s) d'un système simple ( $Z = 1, n = 0$ ), un système utilisant la compression toute seule ( $Z = 1, n = 5$ ), un autre système avec ( $Z = 5, n = 0$ ) et finalement le système ( $Z = 5, n = 5$ ) :

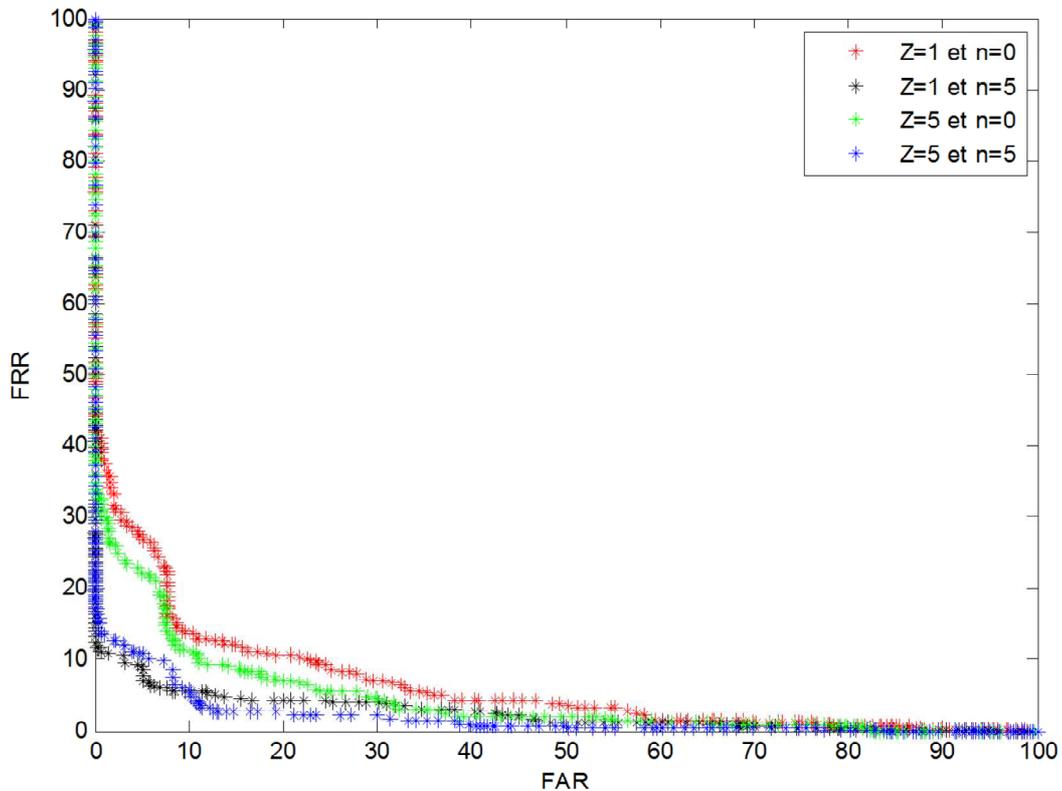


Figure 5-18 : Les courbes ROC(s).

On remarque de la Figure 5-18 que le niveau de compression «  $n$  » améliore plus les performances du système que le paramètre «  $Z$  » (classe de la BDD).

### 7.1.2. Mesure du taux d'identification

Dans cette partie, pour un nombre de classes de 1 à 5, avec un nombre de visages propres égal à 14, les résultats obtenus sont :

Classe de la BDD	1	2	3	4	5
Taux d'identification	90.20	90.85	92.22	92.62	93.43

Tableau 5-7 : Taux d'identification en fonction du nombre d'éléments dans la classe «  $Z$  ».

Il faut aussi noter que lorsque la classe de la BDD est augmentée, les nouvelles images des  $N = 132$  individus présents dans la BDD seront traitées comme des images de visages différents, ce qui donne un nombre de valeurs et visages propres égal à  $Z \times N$ .

D'après le Tableau 5-7 le taux d'identification du système s'améliore lorsque «  $Z$  » augmente car on donne plus d'informations à notre système concernant chaque personne, mais le temps de traitement augmente car le nombre d'éléments avec lequel on calcule les distances euclidiennes est égal au nombre d'éléments de la BDD qui augmente avec  $Z$ .

Cette technique peut être aussi combinée avec celle de compression WPD, le

Tableau 5-8 résume les différents résultats obtenus par la combinaison des deux paramètres  $n$  et  $Z$  (Pour la même BDD et un nombre de visages propres égal à 14)

Classes de la BDD	Niveau de compression <i>n</i>	Taux d'identification
1	1	90.3030
	2	90.7071
	3	91.7172
	4	93.4343
	5	95.8586
2	1	90.9091
	2	91.3131
	3	91.9697
	4	94.2929
	5	96.7172
3	1	92.2727
	2	92.4747
	3	93.0303
	4	95.2525
	5	97.1717
4	1	92.6768
	2	92.8788
	3	93.5354
	4	95.5556
	5	97.3232
5	1	93.4848
	2	93.5354
	3	94.3939
	4	96.0101
	5	<b>97.6768</b>

Tableau 5-8 : Taux d'identification en fonction du nombre « Z » classe d'une image et du niveau de compression « n ».

La combinaison des deux méthodes est très intéressante car elle permet d'atteindre des taux d'identification très remarquables, une simple PCA assure un taux = 90.2020 %, Par contre la combinaison de la méthode PCA et WPD (n=5 et Z=5) assurera un taux = 97.6768 %.

## 7.2. Ajout d'un second seuil

Cette technique est proposée pour augmenter le taux d'identification seulement. Lors de la phase de classification le système peut soit :

- **Bien identifier la personne** : Le système trouve la bonne personne après avoir calculé toutes les distances euclidiennes possibles avec la BDD. Si on classe ces distances de façon croissante, on remarquera que la marge entre la première valeur et la seconde est importante,
- **Faire une erreur d'identification** : Dans ce cas la marge entre le premier et le second élément de l'ensemble des distances euclidiennes est faible.

**Exemple :** Deux images tests de la BDD, une image est reconnue et l'autre non avec le système VP à base de PCA ( $N = 132$ ,  $Z = 1$ ,  $VP = 14$ ,  $n = 1$ ) :

Image de Test	Résultat	Image d'apprentissage	Première distance min	Seconde distance min	La différence entre les deux
	Correct		4.8057	9.3629	4.5572
	Faux		6.7777	8.4421	1.6644

Tableau 5-9 : Le principe du second seuil.

On remarque que la deuxième photo de test du tableau 5-9 n'est pas reconnue, la différence entre la première et la seconde distance euclidienne (de l'ensemble des distances euclidiennes classé par ordre croissant) est très faible. Donc pour éviter ce type d'erreur, on doit imposer une marge minimale « Seuil  $S_m$  » entre le premier et le second élément de l'ensemble des distances euclidiennes.

Si ce seuil n'est pas respecté le système peut procéder à une des solutions suivantes :

- Le rejet direct de la personne,
- L'acquisition d'une nouvelle image de l'utilisateur,
- ...etc.

Cette technique d'amélioration du taux d'identification du système RV peut être ajoutée au système à base de PCA simple ou au système PCA et WPD combinés.

Il faut noter aussi que la valeur de ce seuil est déduite expérimentalement.

La Tableau 5-10 résume quelques résultats obtenus lors de la simulation, pour la programmation avec Matlab. Dans le cas où la condition du second seuil n'est pas vérifiée le système ne prend pas de décision concernant l'image.

Z	1	2	3
Bonne identification	1 322	1 488	1 532
Mauvaise identification	12	9	7
Décision non prise	646	483	441
Taux de prise de décision %	67.4	75.6	77.7
Taux d'identification totale %	66.8	75.2	77.4
Taux d'identification %	99.1	99.4	99.5

Tableau 5-10 : Les résultats obtenus pour un seuil  $S_m=4.9$

On remarque que lorsque  $Z$  augmente le nombre de visages, la non prise de décision par le système ne prend pas de décision diminue et cela avec une amélioration du taux d'identification.

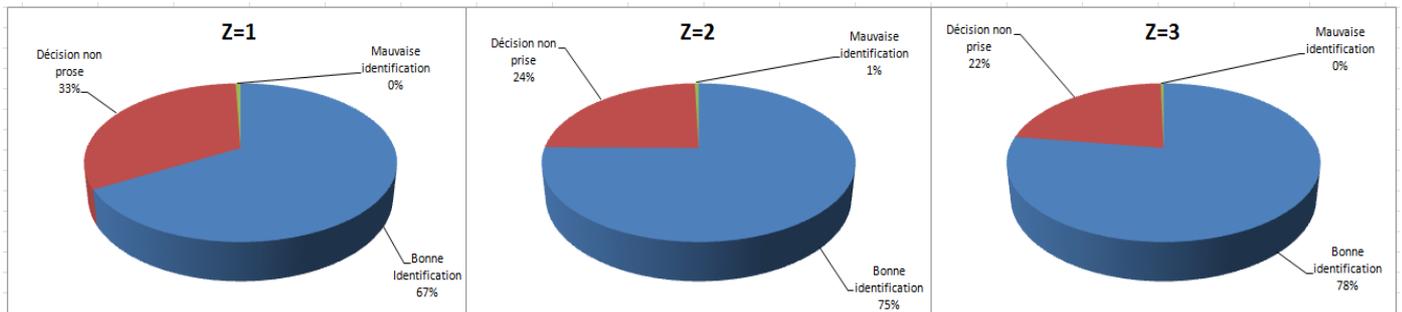


Figure 5-19 : Résultat de la simulation.

## 8. Conclusion :

Les résultats obtenus concernant les différents taux de reconnaissance avec l'algorithme PCA sont moyens et peuvent être améliorés pour atteindre de meilleurs résultats. Pour cela plusieurs techniques peuvent être ajoutées au système RV à base du PCA pour l'améliorer, parmi lesquelles celles déjà citées dans ce PFE :

- La compression d'images avec l'algorithme WPD pour l'élimination de détails inutiles,
- L'augmentation des données utilisées pendant la phase d'apprentissage,
- L'ajout d'un second seuil pour les systèmes d'identification.

Il existe d'autres techniques qui pourraient aussi améliorer la performance du système biométrique comme l'utilisation de la multimodalité (Chapitre 1).

# Chapitre 6 : Implémentation sur FPGA des deux algorithmes et de leur combinaison.

---

## 1. Introduction

---

Dans cette partie on présente les circuits ciblé par notre PFE qui sont les FPGA. La présentation de ces circuits, la description de leur fonctionnement, leurs avantages, ainsi que leur programmation seront donnés. On présentera par la suite le langage VHDL utilisé pour programmer les FPGA.

La dernière partie de ce chapitre résume la façon avec laquelle le problème d'implémentation des deux algorithmes PCA et WPD est résolu, par des schémas blocs et quelques résultats obtenus lors de l'implémentation des algorithmes.

## 2. Les circuits programmables FPGA [38]

---

Grâce au transistor et au traitement numérique des données notre monde est en changement technologique accéléré (GSM, GPRS, GPS, Wifi, Ipod, Iphone, ...). Cette évolution technologique est très rapide telle que personne ne peut prédire quel nouveau système ou produit sera sur le marché les années suivantes.

Parmi, les nouveaux arrivés dans le domaine numérique, les circuits programmables et plus spécialement les circuits FPGA (Field Programmable Gate Array) ne cessent de se tailler une place de choix. Depuis leur invention par Xilinx en 1984, les FPGA sont passés d'un simple rôle d' « interfaçage d'appoint » à celui de remplacer les ASIC (Circuits Intégrés à Application spécifique) et les processeurs personnalisés (DSP) dans des applications de contrôle et de traitement de signaux. Le marché mondial du FPGA à atteint en 2005, 1.9 milliards de dollars ; ce chiffre a augmenté jusqu'à atteindre en 2010, 2.75 milliards de dollars.

### 2.1. Très brève présentation des FPGA

---

L'architecture des FPGA actuels se compose d'un réseau d'éléments logiques de base, complété par un réseau maillé d'interconnexions programmables, avec un grand nombre d'entrées/sorties (plus de 1 000 broches sur certains modèles). La sélection des fonctions logiques, la configuration des interconnexions et la définition des entrées/sorties sont réalisées en téléchargeant un fichier binaire de configuration à l'intérieur du FPGA (Bitstream). Sur certains FPGA, le fichier de configuration est stocké dans une mémoire non-volatile et il est ainsi conservé lorsque le système n'est pas sous tension. Sur d'autres, la configuration est stockée dans la mémoire volatile, et le circuit doit donc être réinitialisé à chaque mise sous tension. Certaines familles de FPGA permettent aussi de configurer juste une partie de toute la logique implantée (reconfiguration partielle) sans arrêt de l'ensemble. Ceci équivaldrait à pouvoir changer une Unité Arithmétique et Logique d'un processeur d'ordinateur sans la moindre interruption, voir changer un cœur du dernier CPU par un autre plus adapté au traitement que nous voulons lui soumettre.

Chaque élément logique se compose typiquement d'une table de vérité qui implémente des fonctions logiques, d'un registre, et d'une logique dédiée pour permettre de réaliser les opérateurs arithmétiques comme par exemple des additionneurs. Les registres sont cadencés par des horloges qui se "propagent" dans le FPGA avec de très faibles variations temporelles. Les innovations récentes de la technologie FPGA ont permis d'améliorer la densité des éléments logiques et les possibilités d'entrées/sorties de données. On trouve couramment aujourd'hui des FPGA comportant des blocs mémoire de 20 Ko; les plus gros FPGA peuvent intégrer jusqu'à 400 de ces blocs.

Les éléments logiques des FPGA peuvent également comporter des multiplieurs, des additionneurs, des accumulateurs, des contrôleurs digitaux d'horloges, des couches MAC pour Ethernet, des PHY-MAC pour PCIExpress, etc. En outre, les FPGA les plus récents comportent également des microprocesseurs embarqués soit sous forme de silicium ou de logique (Hardcore CPU / Softcore CPU).

Si les FPGA rencontrent un grand succès dans plusieurs secteurs, c'est parce qu'ils réunissent le meilleur des ASIC et des systèmes basés processeur. Ainsi, ils offrent un cadencement par matériel qui leur assure vitesse et fiabilité, mais sont plus rentables que les ASIC personnalisés. Les circuits reprogrammables jouissent également de la même souplesse d'exécution logicielle qu'un système basé processeur, mais ils ne sont pas limités par le nombre de cœurs de traitement disponibles. Contrairement aux processeurs, les FPGA sont vraiment parallèles par nature, de sorte que plusieurs opérations de traitement différentes ne se trouvent pas en concurrence pour l'utilisation des ressources. Chaque tâche de traitement indépendante est affectée à une section spécifique du circuit, et peut donc s'exécuter en toute autonomie sans dépendre aucunement des autres blocs logiques. En conséquence, on peut accroître le volume de traitement effectué sans que les performances d'une partie de l'application n'en soient affectées pour autant.

### 2.2. Rôle de l'ingénieur dans le développement FPGA

---

Deux spécialités de l'ingénierie se retrouvent souvent dans le domaine du développement FPGA :

- Les ingénieurs en génie électriques,
- Les ingénieurs en génie informatique.

Le rôle d'un ingénieur en FPGA commence dès la phase de développement d'un produit incorporant cette technologie. Tout d'abord il lui incombe de choisir le circuit le plus adapté en fonction de la taille nécessaire, des ressources et des performances souhaitées. Il doit aussi tenir compte de l'aspect financier dans son choix pour limiter le coût final. Il doit ensuite procéder avec l'ingénieur de conception matériel au choix des différents circuits s'interfaçant avec le FPGA et identifier les contraintes de placement sur le circuit imprimé et de routage des diverses lignes d'interconnexions. Il s'assure aussi que l'alimentation et la fréquence maximale de fonctionnement souhaitées ne seront pas affectées par la montée en charge quand tout le circuit va contenir de la logique. En effet, la consommation de puissance du circuit ainsi que sa fréquence de fonctionnement maximale possible sont fonction de la taille du circuit et de son taux d'occupation. Plus le circuit va être chargé, plus grande sera sa consommation et plus basse sera sa fréquence maximale de fonctionnement.

Une fois la phase de conception terminée, le développement peut commencer en attendant la réception des premiers prototypes. Cette phase consiste à mettre en place toute la logique d'interfaçage avec les circuits externes et le développement des algorithmes de traitement s'il y a lieu. Pour cela, on utilise majoritairement des langages de programmation dédiés au matériel (HDL : hardware description langage). Deux langages se partagent le marché actuellement : le VHDL et le Verilog. Lors de cette phase, le design est décomposé en petites entités qui sont codées et simulées unitairement. On procède aussi à des tests sur des cartes de développement afin de s'assurer du bon fonctionnement dans des conditions réelles. À cette étape, le déverminage se fait soit avec le simulateur soit avec des oscilloscopes et des analyseurs logiques sur des cartes de développement. Il faut noter que pour les algorithmes de traitement de signal, il existe des outils de génération automatique du code HDL à partir d'un haut niveau d'abstraction ou de code C.

La technologie étant rarement stagnante, il est souvent nécessaire de faire des mises à jour du Firmware du FPGA pour régler certains bugs non détectés avant le release du produit, ou tout simplement pour ajouter des fonctionnalités que l'on a mis de côté pour arriver au marché avant la concurrence.

### 3. Pourquoi les FPGA ? [39]

---

Les principales causes de l'utilisation des FPGA dans le développement et la construction de circuits de traitement de signal numérique sont les atouts de la technologie FPGA suivantes :

- Performances,
- Temps de mise sur le marché,
- Coût,
- Fiabilité,
- Maintenance à long terme.

#### 3.1. Performances

---

Comme ils tirent partie du parallélisme matériel, les FPGA offrent une puissance de calcul supérieure à celle des processeurs de signaux numériques (DSP), car ils s'affranchissent du modèle d'exécution séquentielle et exécutent plus d'opérations par cycle d'horloge. BDTI, une importante société d'analyse et de « benchmarking », a publié des études montrant que les FPGA peuvent offrir une puissance de traitement par dollar plusieurs fois supérieure à celle d'une solution DSP dans certaines applications<sup>2</sup>. Contrôler les entrées et sorties (E/S) au niveau matériel permet d'obtenir des temps de réponse plus courts ainsi que des fonctionnalités spécifiques, qui répondent mieux aux besoins de l'application.

#### 3.2. Temps de mise sur le marché

---

Face à des préoccupations croissantes concernant les temps de mise sur le marché, la technologie FPGA représente une solution souple offrant des capacités de prototypage rapide. Ainsi, on peut tester une idée ou un concept, puis le vérifier sur du matériel sans avoir à passer par le long processus de fabrication d'un ASIC personnalisé<sup>3</sup>. Par la suite, on peut apporter les éventuelles modifications nécessaires à notre FPGA, en quelques heures au lieu de quelques semaines. Le matériel « sur étagère » actuellement commercialisé propose également différents types d'E/S déjà connectées à un circuit FPGA programmable par l'utilisateur. La multiplication des outils logiciels de haut niveau disponibles sur le marché permet de réduire le temps d'apprentissage avec les couches d'abstraction. Ces outils comprennent souvent des cœurs de propriété intellectuelle (fonctions précompilées) utiles pour le contrôle avancé et le traitement de signaux.

#### 3.3. Coût

---

Les coûts d'ingénierie non récurrents (NRE) des ASIC personnalisés sont bien supérieurs à ceux des solutions matérielles basées sur du FPGA. L'important investissement de départ que requièrent les ASIC se justifie largement pour les OEM, par exemple, qui peuvent livrer des circuits par milliers chaque année. Cependant, la plupart des utilisateurs finaux ont besoin de matériels personnalisés pour quelques dizaines ou quelques centaines de systèmes en développement. Par nature, les circuits programmables n'impliquent ni coût de fabrication, ni longs délais d'assemblage. Les besoins de la plupart des systèmes évoluent avec le temps ; or la modification progressive d'un FPGA représente un coût négligeable comparé à la dépense considérable qu'exige la reconception d'un ASIC.

#### 3.4. Fiabilité

---

Tandis que les outils logiciels fournissent l'environnement de programmation, les circuits FPGA sont une véritable implémentation matérielle de l'exécution logicielle. Les systèmes basés processeur comprennent souvent plusieurs couches d'abstraction, pour aider à la planification des

---

<sup>2</sup>BDTI Focus Report: FPGAs for DSP, Deuxième Édition, BDTI Benchmarking, 2006.

<sup>3</sup>FPGAs accelerate time to market for industrial designs », M. Thompson, EE Times, 2 juillet 2004, <http://www.us.design-reuse.com/articles/8190/fpgas-accelerate-time-to-market-for-industrial-designs.html>.

tâches et à la répartition des ressources entre les différents processus. La couche de driver contrôle les ressources matérielles et le système d'exploitation gère la mémoire et la bande passante du processeur. Sur chaque cœur de processeur, une seule instruction peut s'exécuter à la fois ; c'est pourquoi les systèmes basés processeur risquent toujours de voir des tâches prioritaires entrer en conflit. Les FPGA, qui n'utilisent pas de système d'exploitation, minimisent les problèmes de fiabilité car ils assurent une exécution véritablement parallèle et un matériel déterministe dédié à chaque tâche.

### 3.5. Maintenance à long terme

Les circuits FPGA sont évolutifs et nous épargnent donc la dépense de temps et d'argent qu'implique la reconception des ASIC. Les spécifications des protocoles de communication numériques, par exemple, évoluent avec le temps. Or les interfaces basées sur ASIC peuvent poser des problèmes de maintenance et de compatibilité. Comme ils sont reconfigurables, les circuits FPGA sont capables de s'adapter aux modifications éventuellement nécessaires. À mesure qu'un produit ou qu'un système évolue, nous pouvons y intégrer des améliorations fonctionnelles sans perdre de temps à reconcevoir le matériel ou à modifier l'implantation du circuit.

## 4. Description du FPGA [41]

Les circuits FPGA ne possèdent pas de programme résident. A chaque mise sous tension, il est nécessaire de les configurer. Leur configuration permet d'établir des interconnexions entre les CLB et IOB. Pour cela, ils disposent d'une RAM interne dans laquelle sera écrit le fichier de configuration. Le format des données du fichier de configuration est produit automatiquement par le logiciel de développement sous forme d'un ensemble de bits organisés en champs de données.

La complexité des composants entraîne le besoin pour les premiers d'avoir des outils de modélisation de haut niveau et de simulation logique rapide (comportementale) adopté rapidement par les seconds.

L'architecture, retenue par Xilinx, se présente sous forme de deux couches :

- Une couche appelée circuit configurable,
- Une couche réseau mémoire SRAM.

La couche dite « *circuit configurable* » est constituée d'une matrice de blocs logiques configurables (CLB) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs entrées/sorties (IOB) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs. La programmation du circuit FPGA appelé aussi LCA (logic cells arrays) consistera par le biais de l'application d'un potentiel adéquat sur la grille de certains transistors à effet de champ à interconnecter les éléments des CLB et des IOB afin de réaliser les fonctions souhaitées et d'assurer la propagation des signaux. Ces potentiels sont tout simplement mémorisés dans le réseau mémoire SRAM.

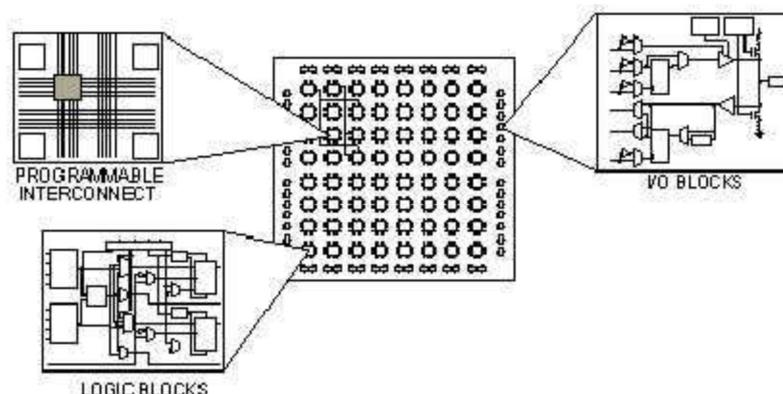


Figure 6-1 : Architecture interne d'un FPGA

Les circuits FPGA du fabricant Xilinx utilisent deux types de cellules de base :

- Les cellules d'entrées/sorties appelés IOB (input output bloc),
- Les cellules logiques appelées CLB (configurable logic bloc). Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable.

#### 4.1. Les CLB (configurable logic bloc)

Les blocs logiques configurables sont les éléments déterminants des performances du FPGA. Chaque bloc est composé d'un bloc de logique combinatoire composé de deux générateurs de fonctions à quatre entrées et d'un bloc de mémorisation/synchronisation composé de deux bascules D. Quatre autres entrées permettent d'effectuer les connexions internes entre les différents éléments du CLB. La Figure 6-2 ci-dessous montre le schéma d'un CLB.

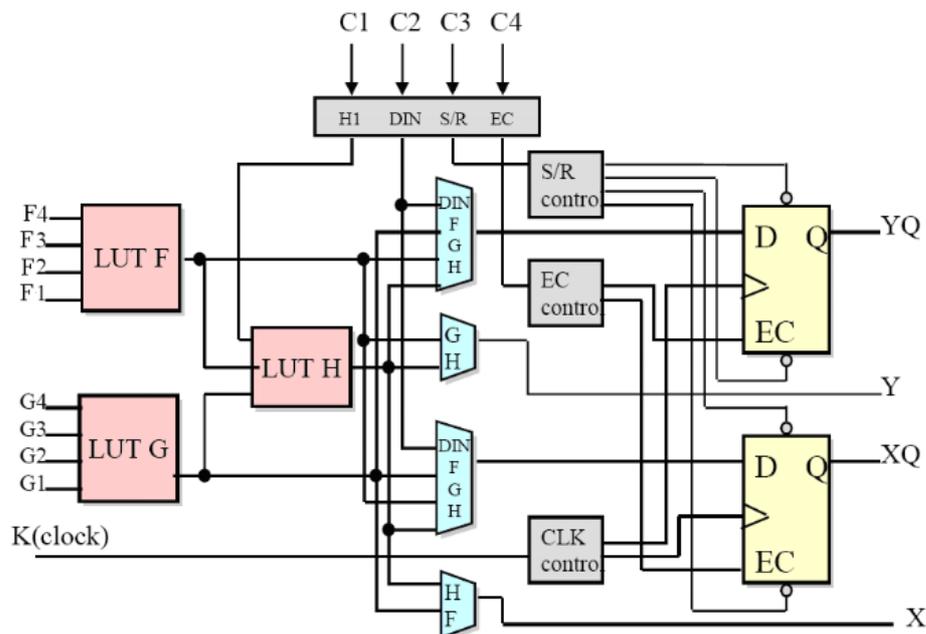


Figure 6-2 : Cellule logique (CLB)

Voyons d'abord le bloc logique combinatoire qui possède deux générateurs de fonctions F' et G' à quatre entrées indépendantes (F1...F4, G1...G4), lesquelles offrent aux concepteurs une flexibilité de développement importante car la majorité des fonctions aléatoires à concevoir n'excède pas quatre variables. Les deux fonctions sont générées à partir d'une table de vérité câblée inscrite dans une zone mémoire, rendant ainsi les délais de propagation pour chaque générateur de fonction indépendants de celle à réaliser. Une troisième fonction H' est réalisée à partir des sorties F' et G' et d'une troisième variable d'entrée H1 sortant d'un bloc composé de quatre signaux de contrôle H1, Din, S/R, Ec. Les signaux des générateurs de fonction peuvent sortir du CLB, soit par la sortie X, pour les fonctions F' et G', soit Y pour les fonctions G' et H'. Ainsi un CLB peut être utilisé pour réaliser:

- deux fonctions indépendantes à quatre entrées indépendantes, plus une troisième fonction de trois variables indépendantes,
- ou toute fonction à cinq variables,
- ou toute fonction à quatre variables et une autre avec quelques fonctions à six variables,
- ou certaines fonctions jusqu'à neuf variables.

L'intégration de fonctions à nombreuses variables diminue le nombre de CLB nécessaire, les délais de propagation des signaux et par conséquent augmente la densité et la vitesse du circuit. Les

sorties de ces blocs logiques peuvent être appliquées à des bascules au nombre de deux ou directement à la sortie du CLB (sorties X et Y). Chaque bascule présente deux modes de fonctionnement : un mode 'flip-flop' avec comme donnée à mémoriser, soit l'une des fonctions F', G', H' soit l'entrée directe DIN et un mode latch. La donnée peut être mémorisée sur un front montant ou descendant de l'horloge (CLK). Les sorties de ces deux bascules correspondent aux sorties du CLB XQ et YQ. Un mode dit de " verrouillage " exploite une entrée S/R qui peut être programmée soit en mode SET, mise à 1 de la bascule, soit en Reset, mise à zéro de la bascule. Ces deux entrées coexistent avec une autre entrée laquelle n'est pas représentée sur la figure, appelée le global Set/Reset. Cette entrée initialise le circuit FPGA à chaque mise sous tension, à chaque configuration, en commandant toutes les bascules au même instant soit à '1', soit à '0'. Elle agit également lors d'un niveau actif sur le fil RESET lequel peut être connecté à n'importe quelle entrée du circuit FPGA.

Un mode optionnel des CLB est la configuration en mémoire RAM de 16x2bits ou 32x1bit ou 16x1bit. Les entrées F1 à F4 et G1 à G4 deviennent des lignes d'adresses sélectionnant une cellule mémoire particulière. La fonctionnalité des signaux de contrôle est modifiée dans cette configuration, les lignes H1, DIN et S/R deviennent respectivement les deux données D0, D1 (RAM 16x2bits) d'entrée et le signal de validation d'écriture WE. Le contenu de la cellule mémoire (D0 et D1) est accessible aux sorties des générateurs de fonctions F' et G'. Ces données peuvent sortir du CLB à travers ses sorties X et Y ou alors en passant par les deux bascules.

#### 4.2. Les IOB (input output bloc)

La Figure 6-3 présente la structure de ce bloc. Ces blocs entrée/sortie permettent l'interface entre les broches du composant FPGA et la logique interne développée à l'intérieur du composant. Ils sont présents sur toute la périphérie du circuit FPGA. Chaque bloc IOB contrôle une broche du composant et il peut être défini en entrée, en sortie, en signaux bidirectionnels ou être inutilisé (haute impédance).

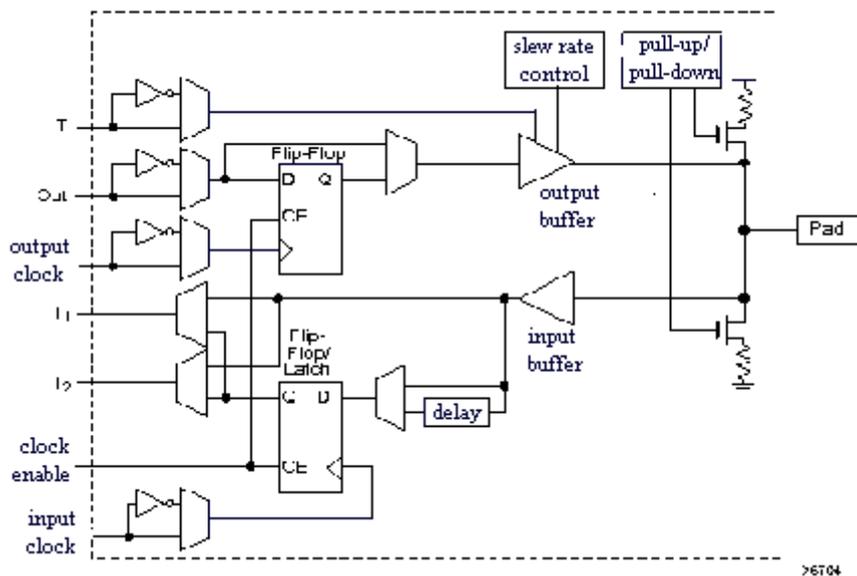


Figure 6-3 : Input Output Block (IOB).

##### 4.2.1. Configuration en entrée

Premièrement, le signal d'entrée traverse un buffer qui selon sa programmation peut détecter soit des seuils TTL, soit des seuils CMOS. Il peut être routé directement sur une entrée directe de la logique du circuit FPGA ou sur une entrée synchronisée. Cette synchronisation est réalisée à l'aide d'une bascule de type D, le changement d'état peut se faire sur un front montant ou descendant. De plus, cette entrée peut être retardée de quelques nanosecondes pour compenser le retard pris par le

signal d'horloge lors de son passage par l'amplificateur. Le choix de la configuration de l'entrée s'effectue grâce à un multiplexeur (program controlled multiplexer). Un bit positionné dans une case mémoire commande ce dernier.

#### 4.2.2. Configuration en sortie

Nous distinguons les possibilités suivantes :

- inversion ou non du signal avant son application à l'IOB,
- synchronisation du signal sur des fronts montants ou descendants d'horloge,
- mise en place d'un " pull-up " ou " pull-down " dans le but de limiter la consommation des entrées/sorties inutilisées,
- signaux en logique trois états ou deux états. Le contrôle de mise en haute impédance et la réalisation des lignes bidirectionnelles sont commandés par le signal de commande Out Enable lequel peut être inversé ou non. Chaque sortie peut délivrer un courant de 12mA. Ainsi toutes ces possibilités permettent au concepteur de connecter au mieux une architecture avec les périphériques extérieurs.

#### 4.3. Les différents types d'interconnexions :

Les connexions internes dans les circuits FPGA sont composées de segments métallisés. Parallèlement à ces lignes, nous trouvons des matrices programmables réparties sur la totalité du circuit, horizontalement et verticalement entre les divers CLB. Elles permettent les connexions entre les diverses lignes, celles-ci sont assurées par des transistors MOS dont l'état est contrôlé par des cellules de mémoire vive ou RAM. Le rôle de ces interconnexions est de relier avec un maximum d'efficacité les blocs logiques et les entrées/sorties afin que le taux d'utilisation dans un circuit donné soit le plus élevé possible. Pour parvenir à cet objectif, Xilinx propose trois sortes d'interconnexions selon la longueur et la destination des liaisons. Nous disposons :

- d'interconnexions à usage général,
- d'interconnexions directes,
- de longues lignes.

##### 4.3.1 Les interconnexions à usage général

Ce système fonctionne en une grille de cinq segments métalliques verticaux et quatre segments horizontaux positionnés entre les rangées et les colonnes de CLB et de l'IOB.

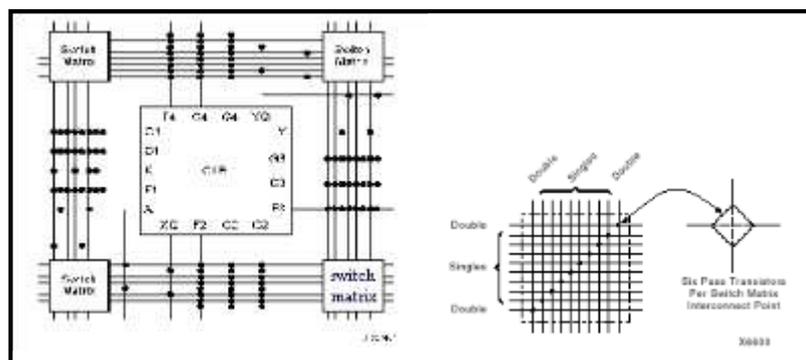


Figure 6-4 : Connexions à usage général et détail d'une matrice de commutation

Des aiguilleurs appelés aussi matrices de commutation sont situés à chaque intersection. Leur rôle est de raccorder les segments entre eux selon diverses configurations, ils assurent ainsi la communication des signaux d'une voie sur l'autre. Ces interconnexions sont utilisées pour relier un

CLB à n'importe quel autre. Pour éviter que les signaux traversant les grandes lignes ne soient affaiblis, nous trouvons généralement des buffers implantés en haut et à droite de chaque matrice de commutation.

#### 4.3.2. Les interconnexions directes [8]

Ces interconnexions permettent l'établissement de liaisons entre les CLB et les IOB avec un maximum d'efficacité en termes de vitesse et d'occupation du circuit. De plus, il est possible de connecter directement certaines entrées d'un CLB aux sorties d'un autre.

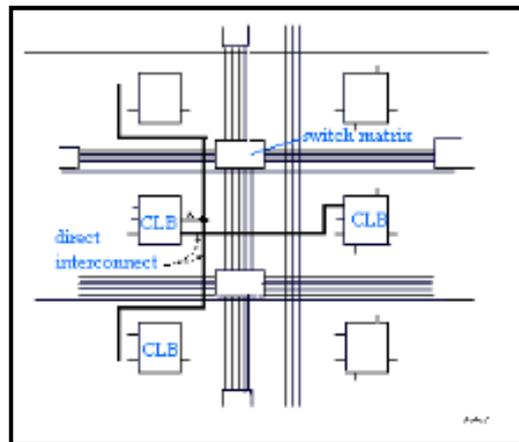


Figure 6-5 : Les interconnexions directes

Pour chaque bloc logique configurable, la sortie X peut être connectée directement aux entrées C ou D du CLB situé au-dessus et les entrées A ou B du CLB situé au-dessous. Quant à la sortie Y, elle peut être connectée à l'entrée B du CLB placé immédiatement à sa droite. Pour chaque bloc logique adjacent à un bloc entrée/sortie, les connexions sont possibles avec les entrées I ou les sorties O suivant leur position sur le circuit.

#### 4.3.3. Les longues lignes

Les longues lignes sont de longs segments métallisés parcourant toute la longueur et la largeur du composant, elles permettent éventuellement de transmettre avec un minimum de retard les signaux entre les différents éléments dans le but d'assurer un synchronisme aussi parfait que possible. De plus, ces longues lignes permettent d'éviter la multiplicité des points d'interconnexion.

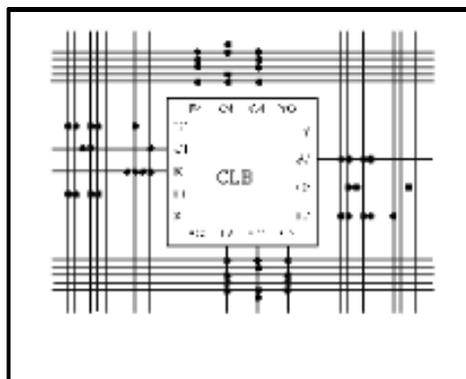


Figure 6-6 : les longues lignes

#### 4.3.4. Performances des interconnexions

Les performances des interconnexions dépendent du type de connexions utilisées. Pour les interconnexions à usage général, les délais générés dépendent du nombre de segments et de la quantité d'aiguilleurs employés. Le délai de propagation de signaux utilisant les connexions directes est minimum pour une connectique de bloc à bloc. Quant aux segments utilisés pour les longues lignes, ils possèdent une faible résistance mais une capacité importante. De plus, si on utilise un aiguilleur, sa résistance s'ajoute à celle existante.

#### 4.4. L'oscillateur à quartz

Placé dans un angle de la puce, il peut être activé lors de la phase de programmation pour réaliser un oscillateur. Il utilise deux IOB voisins, pour réaliser l'oscillateur dont le schéma est présenté ci-dessous. Cet oscillateur ne peut être réalisé que dans un angle de la puce où se trouve l'amplificateur prévu à cet effet. Il est évident que si l'oscillateur n'est pas utilisé, les deux IOB sont utilisables au même titre que les autres IOB.

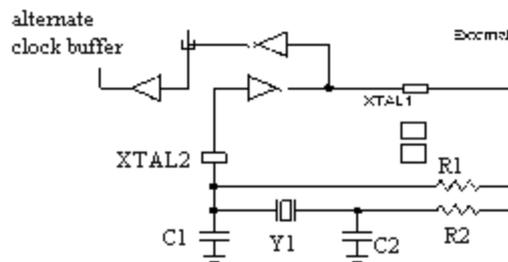


Figure 6-7 L'oscillateur à quartz

## 5. Le langage de description matérielle VHDL [40]

Le VHDL est un langage de description de matériel qui est utilisé pour la spécification (description du fonctionnement), la simulation et la preuve formelle d'équivalence de circuits. Un programme VHDL ne vise pas à une exécution au sens des langages informatiques traditionnels → il ne donne lieu qu'à une exécution par le simulateur.

Son objectif principal : fournir une modélisation hiérarchisée de la fonction à implanter par une description précise de celle-ci.

Ses autres objectifs :

- établir une spécification plus ou moins fine d'une application,
- faire de la preuve formelle,
- établir une documentation précise,
- permettre une synthèse automatique.

**Remarque :** la synthèse électronique est le passage d'un niveau de modélisation de la fonction à son implantation électronique.

### 5.1. Historique

Au début des années 80, Le département de la défense américaine (DOD) désire standardiser un langage de description et de documentation des systèmes matériels ainsi qu'un langage logiciel afin d'avoir une indépendance vis-à-vis de leurs fournisseurs. C'est pourquoi, le DOD a décidé de définir un langage de spécification. Il a ainsi mandaté des sociétés pour établir un langage. Parmi les langages proposés, le DOD a retenu le langage VHDL qui fut ensuite normalisé par IEEE. Cette normalisation a comme objectif :

- La spécification par la description de circuits et de systèmes,
- La simulation afin de vérifier la fonctionnalité du système,
- La conception afin de tester une fonctionnalité identique mais décrites avec des solutions d'implémentation de différents niveaux d'abstraction.

En 1993, une nouvelle normalisation par IEEE du VHDL a permis d'étendre le domaine d'utilisation du VHDL vers :

- La synthèse automatique de circuits à partir des descriptions,
- La vérification des contraintes temporelles,
- La preuve formelle d'équivalence de circuits.

Il existe un autre langage de haut niveau pour la description de matériel. Il s'agit du langage VERILOG. La société cadence est à l'origine de ce langage. Ce langage est aussi normalisé par IEEE.

### 5.2. Pourquoi utiliser le VHDL

Le VHDL est un langage normalisé, cela lui assure une pérennité, Il est indépendant d'un fournisseur d'outils, Il est devenu un standard reconnu par tous les vendeurs d'outils EDA, Cela permet aux industriels d'investir sur un outil qui n'est pas qu'une mode éphémère, c'est un produit commercialement inévitable. Techniquement, il est incontournable car c'est un langage puissant qui permet une excellente lisibilité, une haute modularité et une meilleure productivité des descriptions. Il permet de mettre en œuvre les nouvelles méthodes de conception,

Il est à noter toutefois un inconvénient qui est la complexité du langage. En effet, ce langage s'adresse à des concepteurs de systèmes électroniques, qui n'ont pas forcément de grandes connaissances en langages de programmation.

### 5.3. Ensemble synthétisable en VHDL

L'ensemble du VHDL n'est pas utilisable pour la synthèse automatique. Certaines instructions du VHDL sont clairement non synthétisables, comme par exemple l'instruction « wait for X ns » qui modélise un temps.

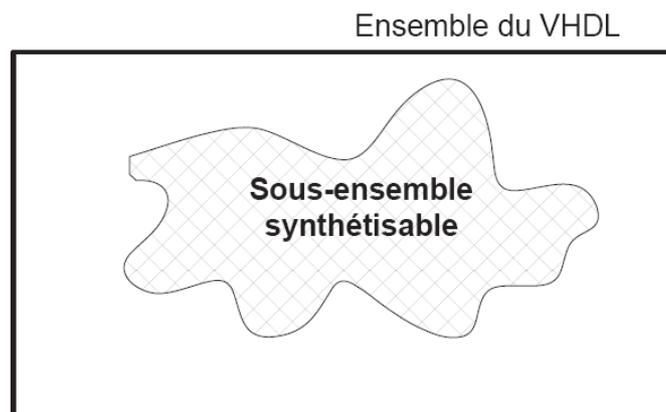


Figure 6-8 : Sous-ensemble synthétisable

La connaissance seule des instructions et de leur syntaxe ne suffit pas. Il faut connaître leur utilisation dans le cadre de la synthèse. A l'inverse, la connaissance de l'ensemble du langage n'est pas nécessaire pour écrire des descriptions synthétisables.

En 1999, IEEE a édité la norme 1076.6 qui définit le sous-ensemble synthétisable. Cette norme définit l'ensemble des syntaxes autorisées en synthèse.

Nous pouvons affirmer que depuis l'année 2000, l'ensemble synthétisable du langage VHDL est de mieux en mieux défini. La majorité des outils de synthèse sont compatibles avec cette norme. Un véritable standard est donc établi pour la synthèse automatique avec le langage VHDL.

#### 5.4. Exemple d'un code VHDL

Le but de cet exemple est de concevoir un module compteur\_décompteur ayant quatre entrées : « reset », « clk » (l'horloge), « clear » (bit de permission) et « u » (bit de choix entre compteur et décompteur) et une sortie « Q » sur quatre bits qui est le résultat du comptage (ou décomptage).

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Counter is
  Port (
    CLOCK : in  STD_LOGIC;
    DIRECTION : in  STD_LOGIC;
    E1,E2,E3: out STD_LOGIC;
    LED : out STD_LOGIC_VECTOR (6 downto 0);
    COUNT_OUT : out  STD_LOGIC_VECTOR (3 downto 0));
end Counter;

architecture Behavioral of Counter is
  signal count_int : std_logic_vector(3 downto 0) := "0000";
begin

  process (CLOCK)
    variable i : integer :=1;
  begin
    if CLOCK='1' and CLOCK'event then
      if DIRECTION='1' then
        count_int <= count_int + 1;
      else
        count_int <= count_int - 1;
      end if;
    end if;
  end process;

end Behavioral;

```

Fichier : « *comp\_decomp.vhd* »

Voici les résultats d'une simulation avec l'éditeur ISE (Xilinx) :

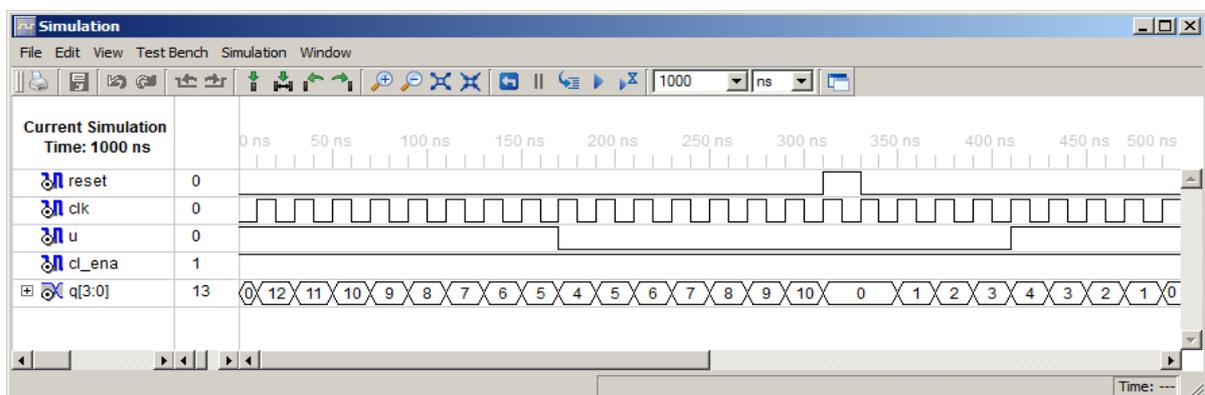


Figure 6-9 : les résultats de notre simulation.

## 6. Les différentes étapes de la synthèse automatique [40]

Le premier outil utilisé dans les étapes de synthèse automatique est un éditeur de texte avec lequel on décrit le comportement du circuit logique en langage VHDL. Ensuite, la description VHDL est traduite par l'outil de synthèse. Cette étape est très importante, il s'agit de convertir la description de haut niveau (VHDL) en fonctions logiques. Le synthétiseur génère en sortie une liste d'interconnexions de composants logiques de base existant dans la technologie cible choisie. Le placeur routeur crée, à partir du fichier généré par le synthétiseur, un fichier spécifique à la technologie cible qui décrit les connexions entre les primitives (composantes élémentaires) à disposition dans le circuit programmable. Finalement, l'intégration physique de la description est effectuée à partir de ce fichier spécifique dans le circuit programmable.

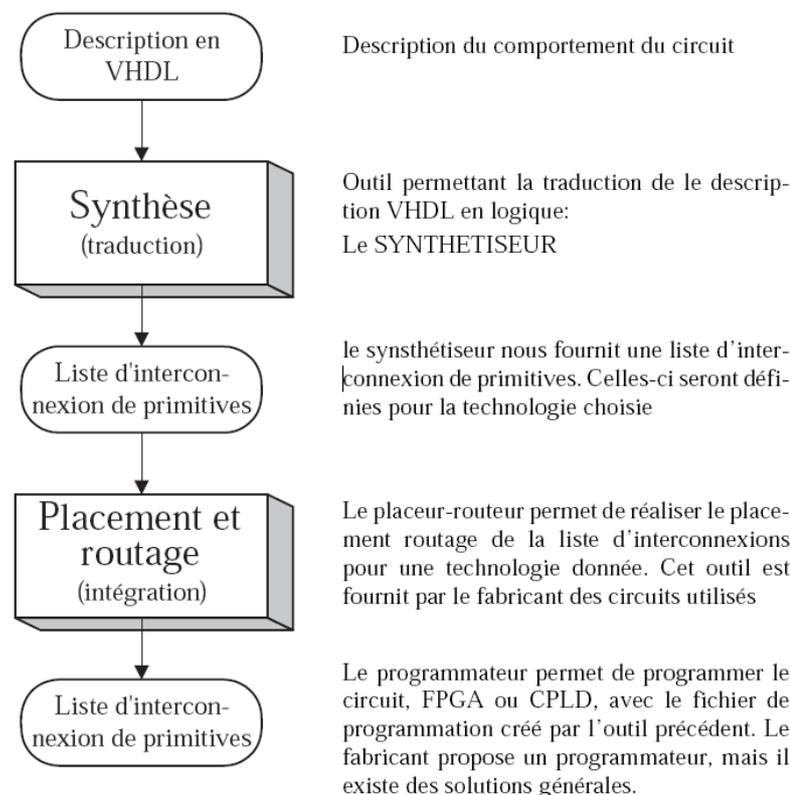


Figure 6-10 : Les étapes de la synthèse automatique.

## 7. L'implémentation des algorithmes PCA, WPD et leur combinaison

Dans cette partie on va voir la façon par laquelle on a implémenté les différents algorithmes utilisés pour le système RV et aussi voir quelques résultats obtenus lors de la simulation des modules à l'aide de l'outil de développement ISE fourni par XILINX.

### 7.1. Analyse en Composantes Principales

Pendant l'implémentation de la PCA, deux méthodes se présentent, la première est l'implémentation de la totalité du système avec ses deux phases principales qui sont la phase d'apprentissage et de classification, la deuxième méthode va implémenter seulement la partie de classification.

L'avantage de la première méthode est que le système réalisé sera universel et pourra fonctionner avec n'importe quelle BDD et être toujours à jour en cas de changements de la BDD (ajout ou suppression de personnes) mais son inconvénient majeur est que l'implémentation de la phase d'apprentissage qui n'est réalisée qu'une seule fois consommera du matériel, contrairement à la seconde méthode où seulement la phase de classification est implémentée, avec l'obtention des

paramètres de la phase 1 (image moyenne, visages propres, poids de la BDD) à l'aide d'un calcul ultérieur effectué par ordinateur.

Dans notre cas, l'image moyenne et les visages propres sont obtenus à l'aide du logiciel Matlab et les poids de la BDD sont calculés avec le module de classification implémenté.

Afin de mieux expliquer l'implémentation de la PCA sur FPGA va suivre les mêmes étapes faites lors de la réalisation du projet nous donnons les étapes suivies dans notre réalisation.

### 7.1.1. Etape n°1

La première étape s'exécute après l'acquisition de l'image et consiste à soustraire l'image moyenne de la BDD de l'image test. Pour pouvoir faire cela l'image moyenne doit être présente dans le bloc, chose qui est réalisée à l'aide d'un bloc ROM1.

- **ROM1** : Ce bloc représente une mémoire de type ROM contenant l'image moyenne de la BDD du système. Le Schéma synoptique de ce bloc est le suivant :

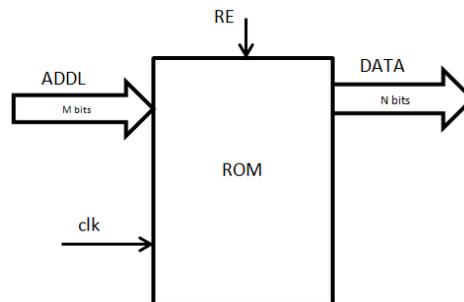


Figure 6-11 : Schéma synoptique de la ROM contenant l'image moyenne.

Où :

- RE : « Read Enable »,
- clk : l'horloge,
- ADDL : Adresse de la donnée,
- DATA : la donnée de l'adresse ADDL.

La soustraction est faite directement pixel par pixel lors de la lecture de l'image test à l'aide de l'horloge qui synchronise la lecture des pixels de l'image test et les pixels de l'image moyenne de la ROM. Pour pouvoir lire tous les pixels présents dans la ROM, un compteur est nécessaire pour permettre d'incrémenter la valeur ADDL et ainsi parcourir tous les pixels de la ROM.

Tout cela est mis dans le bloc CENTRE suivant :

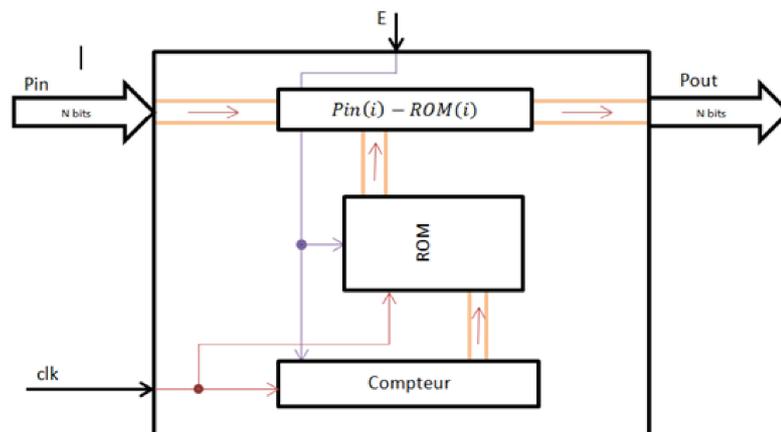


Figure 6-12 : Schéma synoptique du bloc CENTRE

La simulation de ce bloc tout seul avec quelques valeurs seulement et non pas une image complète donne le résultat de la Figure 6-13.

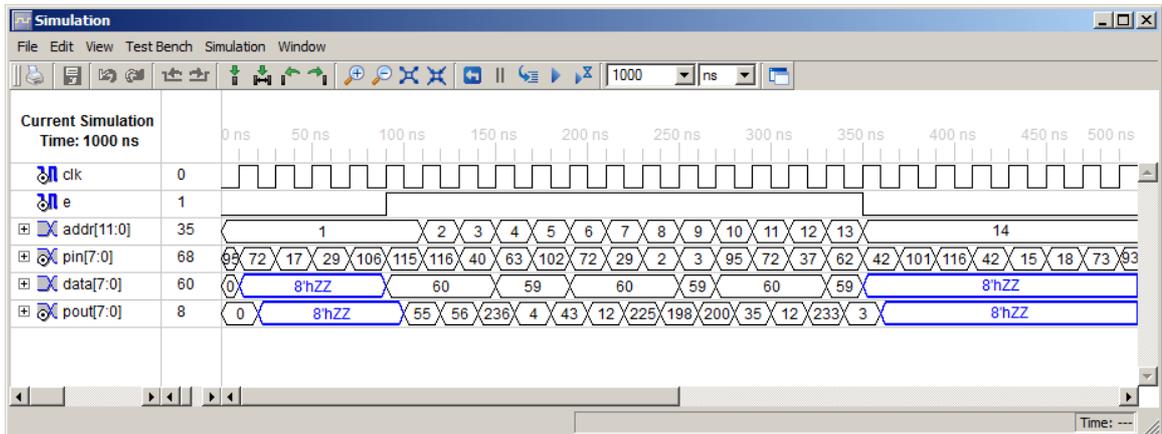


Figure 6-13 : simulation du bloc CENTRE.

### 7.1.2. Etape n°2

Après avoir centré l'image test, il faut maintenant utiliser les résultats du bloc CENTRE pour calculer les poids de l'image test avec les VP de la BDD. Pour l'implémentation de cette partie sur FPGA, on a le choix entre deux méthodes de calcul des poids :

- Séquentielle : un seul bloc pour calculer les «  $P$  » poids de l'image.
- Parallèle : pour calculer les «  $P$  » poids de l'image en même temps on crée «  $P$  » blocs.

La première méthode nous permet de gagner du matériel mais elle n'est pas souhaitable pour des applications temps réel, c'est pour cela qu'on a opté pour la méthode parallèle.

Dans cette partie on doit choisir aussi le nombre de visages propres «  $P$  » à garder pour la génération de l'espace propre. Comme on utilise pour la partie implémentation sur FPGA des BDD contenant seulement 32 visages ou 64 visages, on a choisi donc :  $P = 8$ .

La mesure de un des poids de l'image se fait par une somme de multiplications suivant l'équation suivante :

$$w_i = VP_i^t (\Gamma_{test} - \phi) \tag{6-1}$$

Donc on aura besoin des blocs suivants :

- Une ROM contenant les «  $P$  » visages propres gardés, puisque la méthode parallèle est utilisée, cette ROM devra donc donner en sortie (à chaque front d'horloge) «  $P$  » valeurs correspondant à chaque élément des «  $P$  » visages propres gardés.  
Comme  $P = 8$  donc :

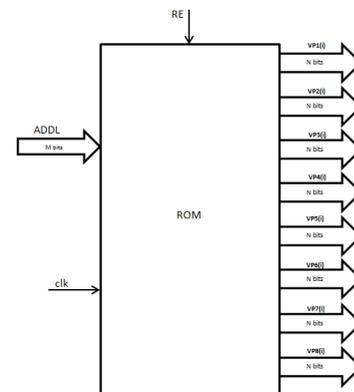


Figure 6-14 : Schéma synoptique de la ROM contenant les  $P=8$  visages propres.

- «  $P = 8$  » blocs MAC pour faire la multiplication et la sommation. Un bloc MAC est constitué d'un simple bloc qui effectue la somme et un autre la multiplication branché de la façon suivante :

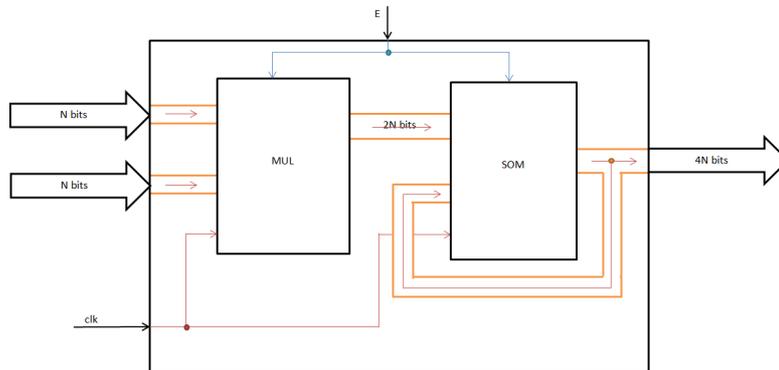


Figure 6-15 : Schéma synoptique d'un bloc MAC.

La simulation avec ce bloc nous donne le résultat suivant :

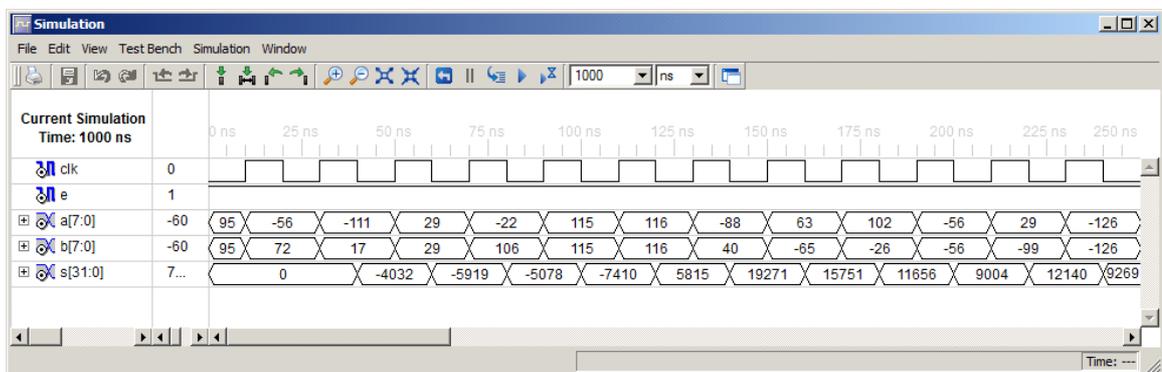


Figure 6-16 : Résultat obtenu par simulation du bloc MAC.

Utilisant l'Erreur ! Source du renvoi introuvable. et le bloc ROM2 et MAC, on construit le loc COORD qui calcule les  $P = 8$  poids des images, la Figure 6-17 montre comment les différents blocs sont reliés. Comme le nombre de pixels des images traitées est fixe, on ajoute donc au système un signal de sortie qui se met à 0 lorsque le système arrive au dernier élément de la ROM2 et le traiter.

Une partie du bloc COORD sera branchée comme suit (pour le calcul d'un seul poids):

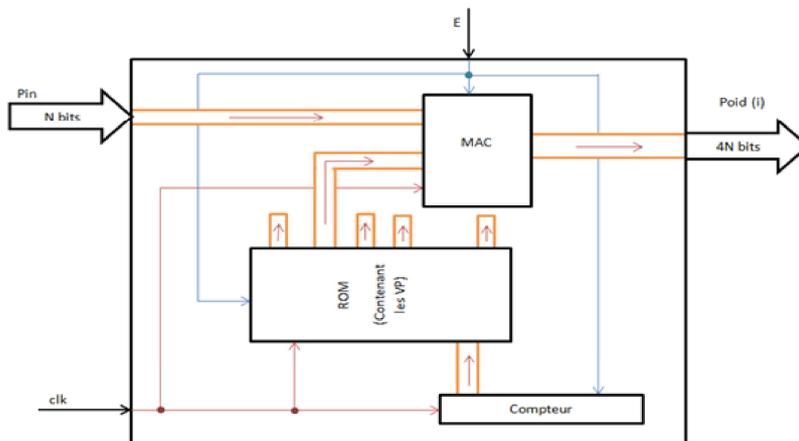


Figure 6-17 : Le bloc COORD.

Après avoir créé le bloc COORD, on peut construire le bloc PART1 qui est constitué des blocs CENTRE et COORD. Le bloc PART1 calcule les poids d'une image de visage de la façon suivante :

- chaque pixel est centré à l'aide du bloc CENTRE en utilisant l'image moyenne qui se trouve dans la ROM1,
- Une multiplication par VP qui est effectuée pixel par pixel jusqu' au dernier pixel de l'image test.

Le bloc PART1 peut être schématisé de la façon suivante :

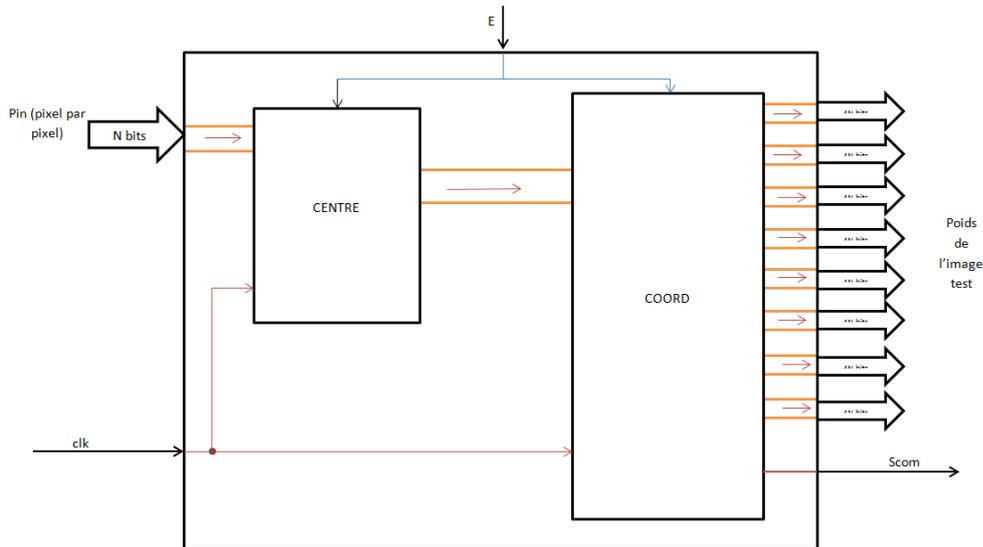


Figure 6-18 : Le bloc PART1.

Pour tester ce dernier composant, on utilise une image de notre BDD en calculant ses poids, la simulation est effectuée avec un « test bench » écrit en code VHDL (qui lit les données a partir d'un fichier « .txt »).

Les fichiers d'extension « .txt » représentant les images sont obtenus à l'aide Matlab qui permet la transformation d'un nombre décimal en un code binaire sur le nombre de bits que l'on désire en utilisant la fonction « quantizer ».

Pour notre application sur FPGA on a choisi de coder les valeurs des pixels en niveaux de gris normalisées sur 8 bits, ce qui donnera des poids de taille égale à 32 bits.

La première BDD utilisée pour la simulation contient 32 images ; ci-dessus, la présentation de la simulation du calcul du poids de la 25<sup>ème</sup> personne donne :

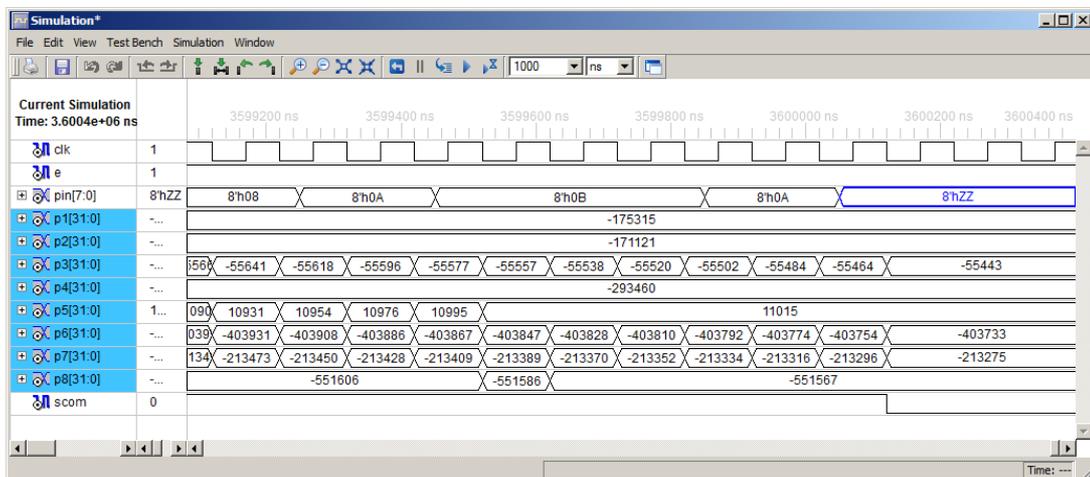


Figure 6-19 : Simulation du bloc PART1.

Dans la Figure 6-19, c'est lorsque le signal « Scom » est nul, qu'on pourra lire les valeurs de sortie comme étant les poids de l'image test.

Donc avec ce premier bloc on calcule tous les poids des éléments présents dans la BDD que l'on a choisi de garder pour pouvoir les utiliser dans le second module.

### 7.1.3. Etape n°3

Après le calcul des poids de l'image test on doit déterminer à quelle personne correspond cette image, pour cela le calcul de similarité adopté est celui de la distance euclidienne.

On mesure toutes les distances euclidiennes entre les poids de l'image test et les poids des images présentes dans la BDD, donc on a besoin d'abord d'une mémoire ROM contenant les poids des images de la BDD ( $N = 32$ ) et d'un bloc qui calcule la distance euclidienne entre deux valeurs, les blocs utilisés sont :

- ROM : Identique à celle utilisée pour les VP, à la différence qu'on a en sortie les 8 poids de l'image se trouvant à l'adresse ADDL,
- Le bloc de mesure de distances euclidiennes est un simple bloc qui calcule la différence élevée au carré de deux valeurs par une simple soustraction et multiplication. Comme on a pour chaque personnes  $P = 8$  poids, on utilisera donc 8 blocs de mesures de distances euclidiennes,
- Les  $P = 8$  valeurs obtenues après la mesure des distances euclidiennes sont sommées pour pouvoir déterminer par la suite le minimum,
- Le bloc qui détermine séquentiellement le minimum des distances euclidiennes.

Avec ces simples composants cités ci-dessus le bloc PART2 peut être réalisé pour nous donner l'indice du minimum lors de l'identification d'une personne, la **Erreur ! Source du renvoi introuvable.** donne le schéma synoptique du bloc PART2 qui rassemble les blocs (ROM : contenant les poids de la BDD,  $P = 8$  blocs pour la mesure des distances euclidiennes, un bloc pour sommer les  $P = 8$  distances, MIN : qui renvoie le minimum d'une série de valeurs ( $N = 32$ )).

Pour terminer la partie du PCA il nous suffit d'assembler les deux blocs PART1 et PART2 ce qui donne :

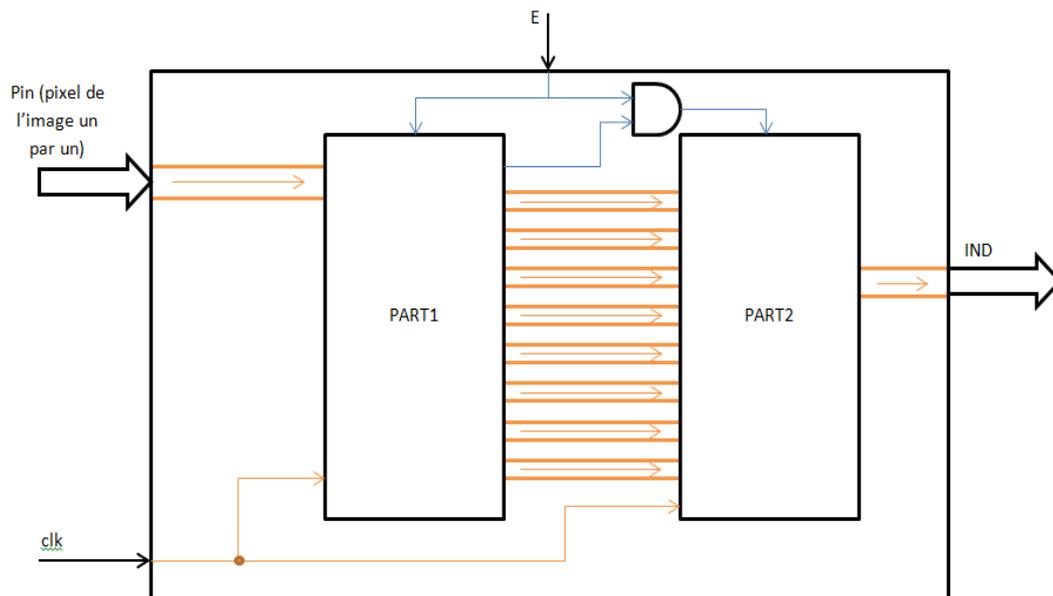


Figure 6-20 : Le bloc ACP avec  $P=8$  VP.

Dans la partie implémentation, on a choisi une BDD de 32 personnes où chacun à une seule image qui le représente, donc l'indice de sortie de notre bloc PCA varie entre 0 et 31 où 0 correspond à la valeur 32, avec l'utilisation aussi d'un seuil qui permettra de rejeter des imposteurs dans le cas où la valeur du minimum est supérieure à ce dernier, en mettant la sortie IND en haute impédance « Z ».

Voici quelques résultats obtenus après l'implémentation de tout le système pour des images de la BDD de taille 200 × 180 pixels:

- Si on prend une image utilisée dans la phase d'apprentissage (personne 8 par exemple) on trouvera que la distance euclidienne minimum sera égale à zéro et que la personne est bien identifiée :

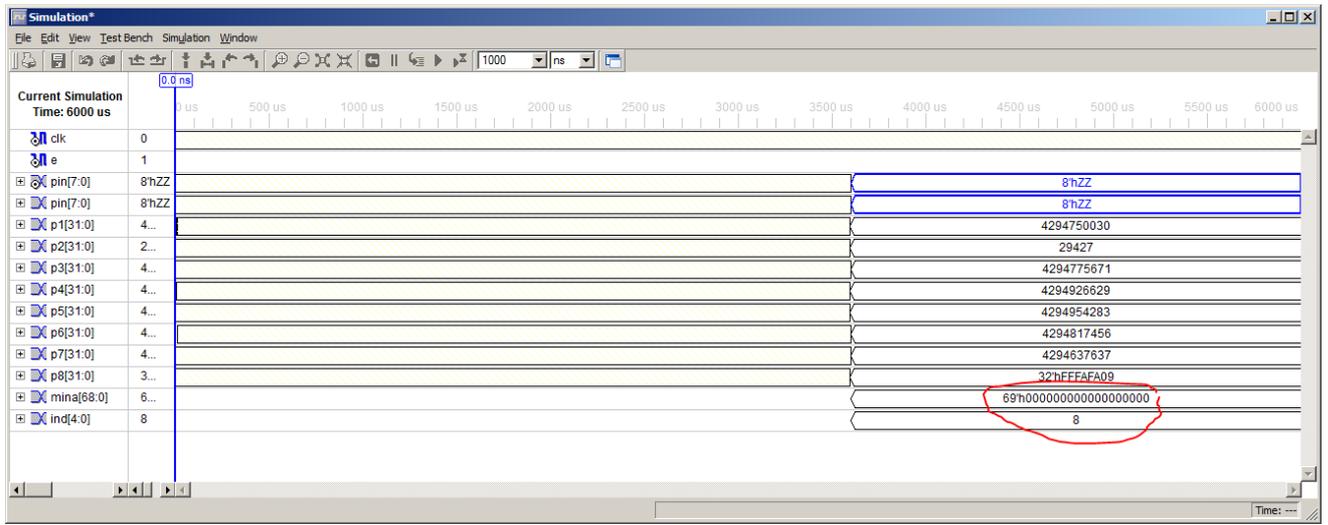


Figure 6-21 : Résultat de simulation avec une image utilisée dans la phase d'apprentissage.

- L'image test est celle de la personne 26 et l'indice 15, on obtient :

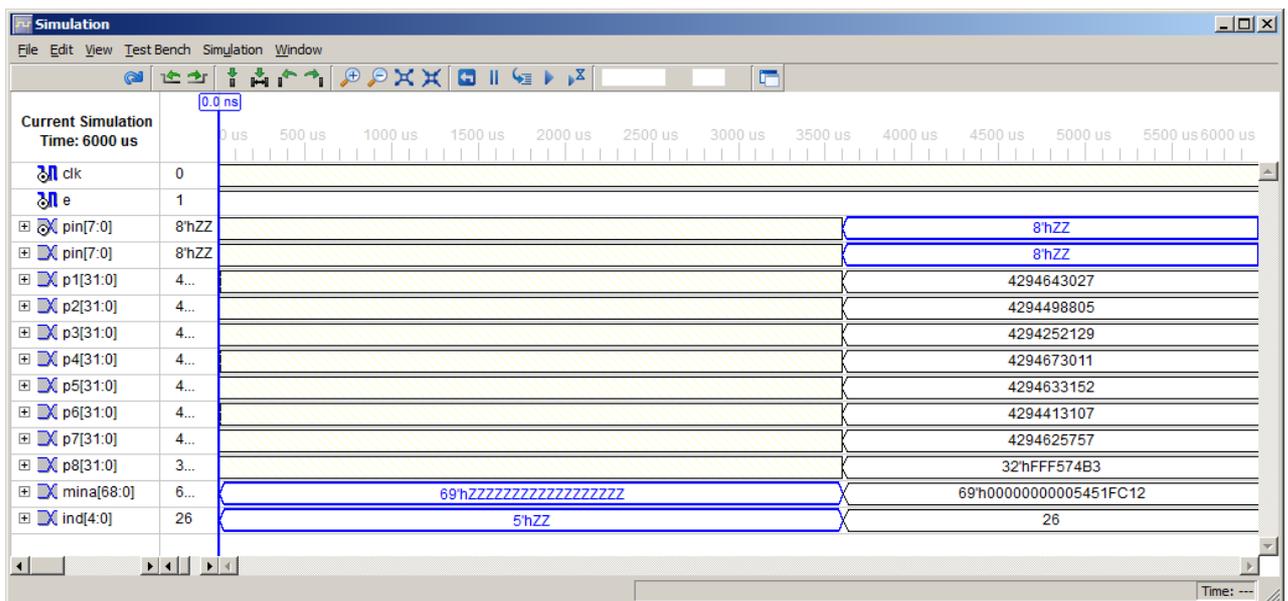


Figure 6-22 : Résultat de simulation avec l'image 15 de la personne 26.

- Image d'imposteur :

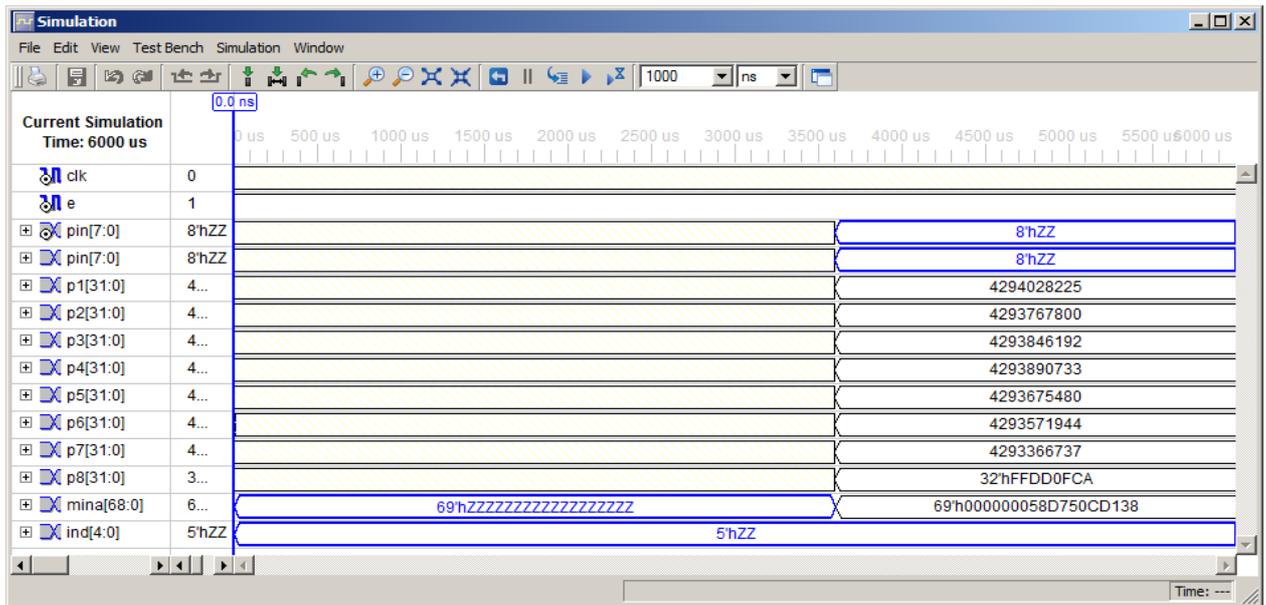


Figure 6-23 : Résultat obtenu avec une image d'un imposteur.

Le seuil est toujours déterminé expérimentalement avec un ensemble d'essais pour atteindre un  $FAR \cong FRR$  comme c'est déjà fait dans la partie simulation sur Matlab.

## 7.2. Décomposition en Paquet d'Ondelettes

La réalisation du module WPD est d'abord effectuée pour le système où le niveau de décomposition est égal à 1 ensuite les autres niveaux peuvent être construits directement par l'utilisation de plusieurs blocs WPD1 l'un après l'autre (en cascade), de cette façon on a fait la WPD2.

Commençons d'abord par WPD1, qui est réalisée par plusieurs processus qui s'exécutent en concurrence et qui réalise les tâches suivantes :

- Lecture de deux colonnes de l'image,
- Lorsqu'on arrive au second élément de la deuxième colonne, le processus qui calcule la moyenne de deux colonnes commence jusqu'à atteindre la limite de la première colonne  $L$ ,
- Le calcul des moyennes des lignes commence lorsque celui des colonnes arrive à la position  $\frac{L}{2} + 2$ .

Ce qui donne finalement en sortie, l'image après décomposition. Mais comme la sortie n'est pas toujours présente à chaque front d'horloge comme l'entrée, un signal «  $I$  » est mis en sortie et vaut :

- « 1 » lorsque la sortie est présente.
- « 0 » lorsque la sortie n'est pas présente.

Voici quelques exemples de l'utilisation de ce bloc avec des données simples et aussi sur une image :

- Ensemble de données (longueur d'une colonne = 8) :

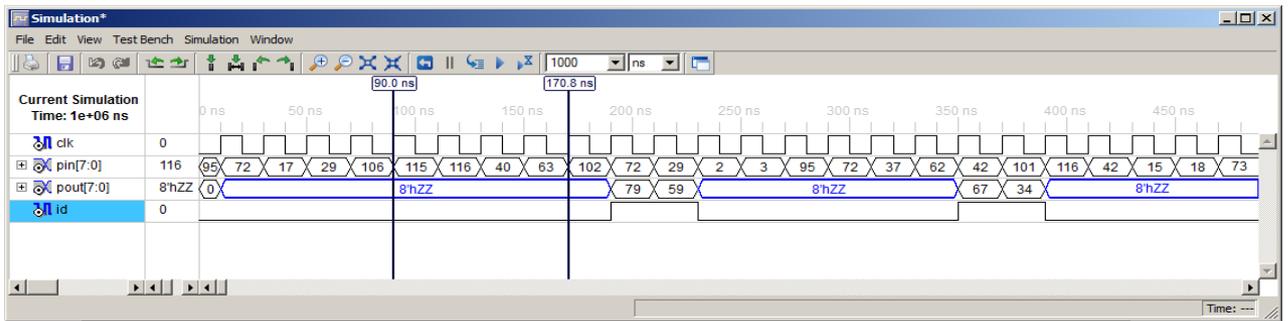


Figure 6-24 : Application de la WPD sur un simple exemple.

- La simulation avec une image : comme pour PCA, notre image est transformée d'abord en un fichier d'extension « .txt » contenant les valeurs de chaque pixels, donc on lit le fichier « o1.txt » et l'image compressée résultante sera elle aussi mise dans un fichier « .txt » puis en utilisant Matlab on fait l'opération inverse pour reconstituer l'image compressée.



Figure 6-25 : Résultat de compression d'image avec ISE.

**La WPD2 :** Ce bloc est constitué à partir de deux blocs WPD1 qui seront reliés de la façon suivante (Figure 6-26) :

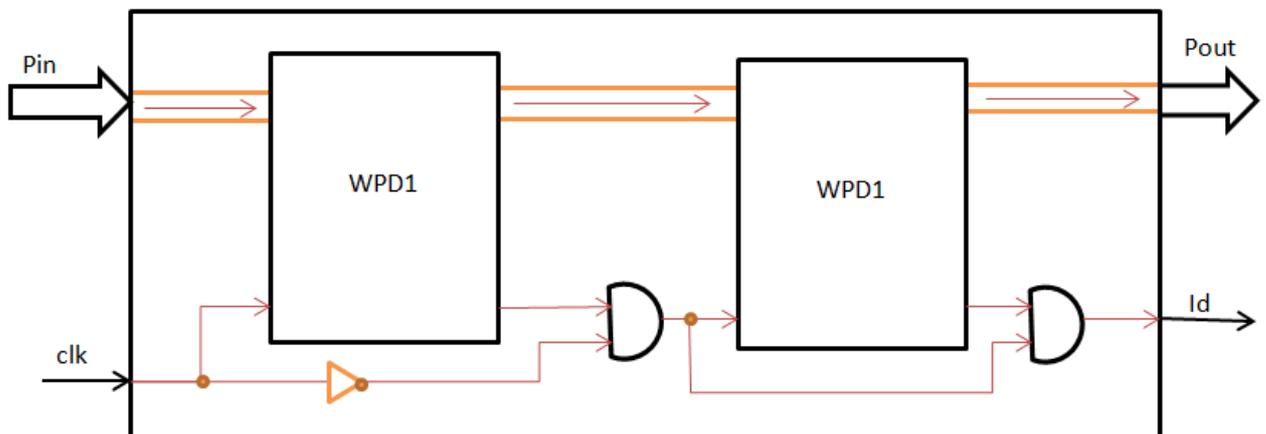


Figure 6-26 : Le bloc WPD2.

Les résultats suivants ont été obtenus :

- Pour une image :



Figure 6-27 : Résultat de la compression avec WPD2.

### 7.3. PCA et WPD

Comme déjà vu dans la partie MATLAB, la combinaison de WPD et PCA améliore le taux d'identification et aussi les taux FAR et FRR du système, cette combinaison permet aussi de rendre le système plus léger en réduisant la taille des différentes images manipulées.

La réalisation du bloc PCAWPD est fait par utilisation des blocs PCA et WPD de la manière suivante :

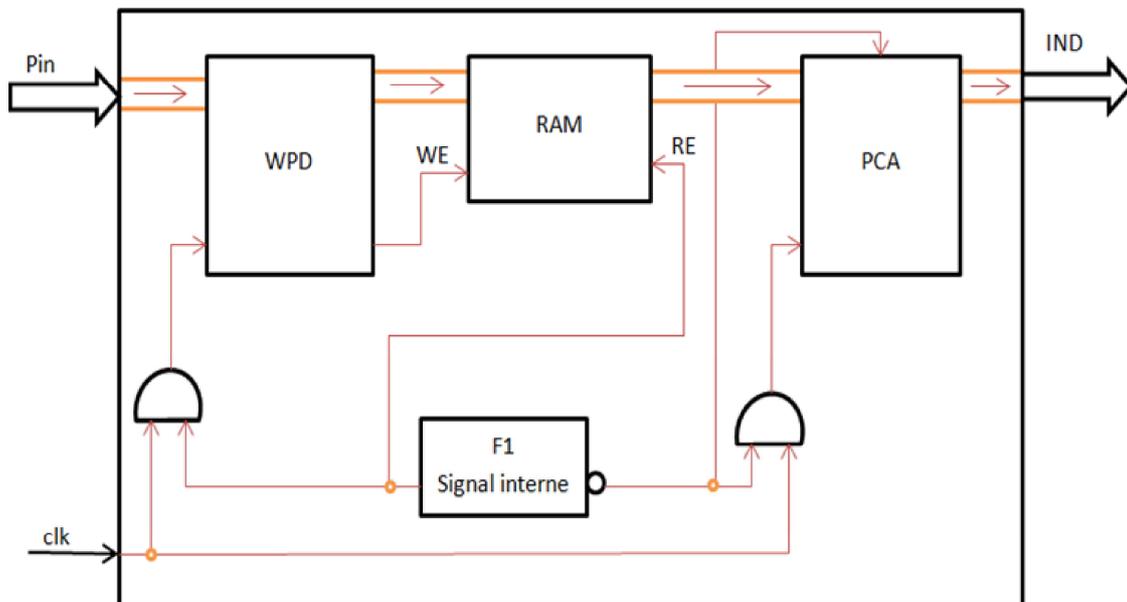


Figure 6-28 : Le bloc PCA\_WPD.



### 8.1. Synthétisation du bloc PCA :

La synthétisation du bloc PCA est faite avec les paramètres suivants :

- Une BDD contenant 32 personnes.
- Un nombre de VP gardé égales à 8.
- Des images de taille (50 × 45) pixels, ce qui correspond à un niveau de compression « n » (En utilisant WPD) égale à « 2 ».

donne :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
Number of Slice Registers:          1898 out of 64000    2%
Number of Slice LUTs:              26941 out of 64000    42%
    Number used as Logic:          26941 out of 64000    42%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 27432
    Number with an unused Flip Flop: 25534 out of 27432    93%
    Number with an unused LUT:      491 out of 27432     1%
    Number of fully used LUT-FF pairs: 1407 out of 27432    5%
    Number of unique control sets:   16

IO Utilization:
Number of IOs:                      84
Number of bonded IOBs:              84 out of 680     12%

Specific Feature Utilization:
Number of Block RAM/FIFO:           11 out of 228     4%
    Number using Block RAM only:     11
Number of BUFG/BUFGCTRLs:          1 out of 32      3%
Number of DSP48Es:                  8 out of 256     3%
    
```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

    Minimum period: 41.010ns (Maximum Frequency: 24.384MHz)
    Minimum input arrival time before clock: 2.121ns
    Maximum output required time after clock: 2.844ns
    Maximum combinational path delay: No path found
    
```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (50 \times 45 + 32)T_{min} = 93,584 \mu s$$

qui est temps pris pour reconnaître une personne de la BDD constitué de 32 personnes.

Si on augmente le nombre de personnes de la BDD, le nombre de VP ou on utilise des images de taille plus grande, on consommera plus de slices et le temps du traitement augmentera.

Dans le cas où on augmente le nombre de personnes de la BDD jusqu'à 64 personnes en gardant toujours la même taille des images et avec 8 VP, on obtient les résultats suivants :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
Number of Slice Registers:          1885 out of 64000    2%
Number of Slice LUTs:              26929 out of 64000    42%
    Number used as Logic:          26929 out of 64000    42%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 27417
    Number with an unused Flip Flop: 25532 out of 27417    93%
    Number with an unused LUT:      488 out of 27417     1%
    Number of fully used LUT-FF pairs: 1397 out of 27417    5%
    Number of unique control sets:   16

IO Utilization:
Number of IOs:                      85
Number of bonded IOBs:              85 out of 680     12%

Specific Feature Utilization:
Number of Block RAM/FIFO:           10 out of 228     4%
    Number using Block RAM only:    10
Number of BUFG/BUFGCTRLs:          1 out of 32      3%
Number of DSP48Es:                  8 out of 256     3%
    
```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

    Minimum period: 41.010ns (Maximum Frequency: 24.384MHz)
    Minimum input arrival time before clock: 2.121ns
    Maximum output required time after clock: 2.849ns
    Maximum combinational path delay: No path found
    
```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (50 \times 45 + 64)T_{min} = 94,897 \mu s$$

## 8.2. Synthétisation du bloc WPD :

Voici les résultats obtenus lors de la synthétisation des blocs WPD1 et WPD2 traitant des images de taille  $200 \times 180$  pixels.

a) WPD1 :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
Number of Slice Registers:          4989 out of 64000    7%
Number of Slice LUTs:              4815 out of 64000    7%
    Number used as Logic:          4815 out of 64000    7%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 9595
    Number with an unused Flip Flop: 4606 out of 9595    48%
    Number with an unused LUT:      4780 out of 9595    49%
    Number of fully used LUT-FF pairs: 209 out of 9595    2%
    Number of unique control sets:   610

IO Utilization:
Number of IOs:                      18
Number of bonded IOBs:              18 out of 680    2%

Specific Feature Utilization:
Number of BUFG/BUFGCTRLs:          1 out of 32    3%

```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

    Minimum period: 6.639ns (Maximum Frequency: 150.621MHz)
    Minimum input arrival time before clock: 3.279ns
    Maximum output required time after clock: 2.858ns
    Maximum combinational path delay: No path found

```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (200 \times 180) T_{min} = 239,004 \mu s$$

b) WPD2 :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
  Number of Slice Registers:          7578 out of 64000  11%
  Number of Slice LUTs:              8436 out of 64000  13%
    Number used as Logic:            8436 out of 64000  13%

Slice Logic Distribution:
  Number of LUT Flip Flop pairs used: 14870
    Number with an unused Flip Flop:  7292 out of 14870  49%
    Number with an unused LUT:        6434 out of 14870  43%
    Number of fully used LUT-FF pairs: 1144 out of 14870   7%
    Number of unique control sets:     838

IO Utilization:
  Number of IOs:                      18
  Number of bonded IOBs:              18 out of 680    2%

Specific Feature Utilization:
  Number of BUFG/BUFGCTRLs:           2 out of 32     6%
    
```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

  Minimum period: 6.574ns (Maximum Frequency: 152.120MHz)
  Minimum input arrival time before clock: 3.353ns
  Maximum output required time after clock: 3.437ns
  Maximum combinational path delay: 3.929ns
    
```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (200 \times 180) T_{min} = 236,664\mu s$$

### 8.3. Synthétisation du bloc PCA\_WPD :

Dans cette partie on a combiné le bloc PCA avec le bloc WPD1 et WPD2 pour traiter des images de taille (100 × 90) pixels avec une BDD contenant 32 personnes, on a obtenues les résultats suivants :

a) PCA\_WPD1 :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
Number of Slice Registers:          4587 out of 64000    7%
Number of Slice LUTs:              34695 out of 64000   54%
    Number used as Logic:          34695 out of 64000   54%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 37467
    Number with an unused Flip Flop: 32880 out of 37467  87%
    Number with an unused LUT:      2772 out of 37467   7%
    Number of fully used LUT-FF pairs: 1815 out of 37467  4%
    Number of unique control sets:   347

IO Utilization:
Number of IOs:                      84
Number of bonded IOBs:              83 out of 680    12%

Specific Feature Utilization:
Number of Block RAM/FIFO:           10 out of 228    4%
    Number using Block RAM only:     10
Number of BUFG/BUFGCTRLs:          4 out of 32    12%
Number of DSP48Es:                  8 out of 256    3%
    
```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

    Minimum period: 40.955ns (Maximum Frequency: 24.417MHz)
    Minimum input arrival time before clock: 3.771ns
    Maximum output required time after clock: 2.844ns
    Maximum combinational path delay: No path found
    
```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (100 \times 90 + 50 \times 45 + 32) T_{min} = 462,054 \mu s$$

b) PCA\_WPD2 :

- Le nombre de Slices, IOB,... utilisés :

```

Slice Logic Utilization:
Number of Slice Registers:          5976 out of 64000    9%
Number of Slice LUTs:              29585 out of 64000   46%
    Number used as Logic:          29585 out of 64000   46%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 33675
    Number with an unused Flip Flop: 27699 out of 33675   82%
    Number with an unused LUT:      4090 out of 33675   12%
    Number of fully used LUT-FF pairs: 1886 out of 33675   5%
    Number of unique control sets:   492

IO Utilization:
Number of IOs:                      84
Number of bonded IOBs:              83 out of 680    12%

Specific Feature Utilization:
Number of Block RAM/FIFO:           9 out of 228     3%
    Number using Block RAM only:     9
Number of BUFG/BUFGCTRLs:          5 out of 32     15%
Number of DSP48Es:                  8 out of 256     3%
    
```

- La fréquence maximale et la période minimale :

```

Timing Summary:
-----
Speed Grade: -2

Minimum period: 40.955ns (Maximum Frequency: 24.417MHz)
Minimum input arrival time before clock: 3.836ns
Maximum output required time after clock: 2.844ns
Maximum combinational path delay: No path found
    
```

Dans ce cas le temps du traitement d'une image de visage sera égale à :

$$temps = (100 \times 90 + 25 \times 23 + 32) T_{min} = 393,454 \mu s$$

## 9. Conclusion

Les résultats obtenus après synthèse des différents blocs sont très promoteurs car ils permettent de réaliser des systèmes de reconnaissance de visages fonctionnant en temps réel pour des BDD contenant un nombre de personnes limite.

La combinaison du bloc PCA et WPD permet de :

- Améliorer les taux de reconnaissances,
- Augmenter le nombre de personnes de la BDD,
- Diminuer le temps de la phase de reconnaissance.

## Conclusion Générale

---

Le choix de la méthode de reconnaissance de visages pour ce PFE a été fait car elle présente les avantages suivants :

- *Pas gênante* : Cette méthode n'oblige pas les utilisateurs de s'arrêter et de s'approcher d'un appareil de mesure, cette mesure peut être faite à distance,
- *Temps d'acquisition faible* : le temps de mesure des caractéristiques biométriques de cette méthode n'est pas important, d'ailleurs elle peut être utilisée même dans une autoroute pour la surveillance des individus,
- *Efficace* : Ce n'est pas la méthode qui donne le meilleur taux d'identification, mais elle possède un taux satisfaisant.

L'objectif principal de notre travail de PFE est l'amélioration des deux paramètres : taux d'identification et temps de traitement.

La méthode globale de reconnaissance de visages est utilisée dans notre travail, en considérant en premier lieu, uniquement l'algorithme PCA, puis en second une combinaison séquentielle des algorithmes WPD et PCA.

L'algorithme PCA est utilisé pour l'extraction des éléments de décision d'une image. En gardant uniquement 10% des visages propres, les résultats d'identifications obtenus en utilisant uniquement cet algorithme sont de l'ordre de 90% pour la BDD utilisée dans notre travail.

La combinaison séquentielle des algorithmes WPD et PCA est essentiellement pour : la compression des images, la réduction du temps de traitement et l'amélioration des performances du système biométrique, en supprimant les détails inutiles pour la décision.

En analysant les résultats obtenus dans la partie "*Simulation en Matlab des deux algorithmes et de leur combinaison*", on voit bien qu'effectivement pour le même nombre de visages propres gardés, l'algorithme WPD a permis d'améliorer ce taux jusqu'à 95.85%. On voit bien donc l'apport de l'algorithme WPD concernant le taux d'identification. On a constaté également durant nos simulations que le temps de traitement a été considérablement réduit.

Pour améliorer encore le temps de traitement, on a opté pour une solution matérielle dédiée, soit une carte FPGA. L'utilisation de ce genre de matériel est pris essentiellement pour la facilité de leur programmation, leur flexibilité et le temps de traitement réduit (de l'ordre du temps réel) qu'ils offrent.

Pendant nos simulations, nous avons remarqué que l'utilisation des FPGA donne presque les mêmes taux d'identifications qu'une solution logicielle, avec un temps de traitement beaucoup plus réduit.

On a donc conclu que, pour une performance maximale pour la méthode globale de reconnaissance de visages, il faut opter pour une combinaison séquentielle des algorithmes WPD et PCA, et utiliser une solution matérielle dédiée pour avoir un temps de traitement de l'ordre du temps réel.

Comme amélioration à ce travail, nous proposons l'utilisation de la multimodalité, soit :

- En combinant ce travail avec d'autres algorithmes de reconnaissance globale de visages (LDA, ICA, ...),

- En combinant ce travail avec d'autres techniques biométriques (La reconnaissance de l'iris, de la rétine, de l'empreinte digitale ...),
- En ajoutant à ce travail un bloc de prétraitement des images pour éliminer les bruits qui influent sur les performances du système biométrique.

# Bibliographie

---

- [1] Alain Boucher « Cours de traitement d'images ». [http://www2.ifi.auf.org/personnel/Alain.Boucher/cours/traitement\\_images/index.html](http://www2.ifi.auf.org/personnel/Alain.Boucher/cours/traitement_images/index.html)
- [2] A. Manzanera ENSTA/UEI « Cours de traitement d'images ». <http://www.ensta.fr/~manzaner/>
- [3] J-P. Cocquerez et S. Philipp « Analyse d'images : filtrage et segmentation ». Edition Masson.
- [4] YahyaZoubir Bahia « Segmentation de l'arbre coronaire à partir de la coronarographie ». Mémoire de magister.
- [5] Xavier Philippeau « Segmentation en régions ».  
».<http://xphilipp.developpez.com/articles/segmentation/regions/>
- [6] E.N.S.B.A.N.A « Initiation au traitement d'images avec Matlab ». [http://www.isir.upmc.fr/UserFiles/File/clady\\_homepage/EPU/matlab\\_ti.pdf](http://www.isir.upmc.fr/UserFiles/File/clady_homepage/EPU/matlab_ti.pdf)
- [7] Nicolas MORIZET « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris ». Thèse présentée pour obtenir le grade de Docteur.
- [8] Florent PERRONNIN, Jean-Luc DUGELAY « Introduction à la Biométrie Authentification des Individus par Traitement Audio-Vidéo ». Institut Eurécom.
- [9] Article de : <http://www.securiteinfo.com> « La biométrie »  
<http://www.securiteinfo.com/conseils/biometrie.shtml>
- [10] <http://www.biometrie-online.net/>
- [11] PYC - ENSERG - 27/09/02 introduction au traitement d'image.
- [12] Wikipedia, « Les empreintes digitales », Site web, <http://fr.wikipedia.org/wiki/Biometrie>
- [13] Wikipedia, « Empreinte Digitale », Site web, [http://fr.wikipedia.org/wiki/Empreinte\\_digitale](http://fr.wikipedia.org/wiki/Empreinte_digitale)
- [14] Didier GUILLERM, « Biometrie Online », Site web, <http://www.biometrie-online.net/technologies/les-modalites>
- [15] J. Daugman, « High Confidence Visual Recognition of Persons by a Test of Statistical Independence », IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, p. 1148-1161, 1993.
- [16] Linternaute.com, « Biométrie », Site Web, <http://www.linternaute.com/science/biologie/dossiers/06/0607-biometrie/iris.shtml>
- [17] <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/fgarac>
- [18] Reda JOURANI, “ Reconnaissance de visages “, mémoire fin d'études, Université Mohammed V-Agdal, Faculté des Sciences – Rabat, 2006.
- [19] C. Valens, A Really Friendly Guide to Wavelets, 1999 c.valens@mindless.com
- [20] Houria AIT ABDESSLAM, Mémoire de magister « Réalisation d'un système d'aide à la détection des aberrations chromosomiques » Février 2005 ENP.
- [21] Michel Misiti, Yves Misiti, George Oppenheim and Jean-Michel Poggi, 'Les ondelettes et leurs applications', Ed. Lavoisier, ISBN 2-7462-0526-2, 2003.

- [22] Victor Racine, Mémoire de fin d'études, Institut Curie, UMR144-CNRS, Compartimentation et dynamique cellulaires, Service d'Imagerie, Mars-Septembre 2002.
- [23] Burrus, C. S. and R. A. Gopinath, H. Guo. Introduction to wavelets and wavelet transforms, A PRIMER. Upper Saddle River, NJ (USA): Prentice Hall, 1998.
- [24] Sheng, Y. Wavelet transform. In: The transforms and applications handbook. Ed. by A. D. Poulakas. P. 747-827. Boca Raton, FL (USA): CRC Press, 1996. The Electrical Engineering Handbook Series.
- [25] [http://www.ai.polymtl.ca/cohen/ELE6812/Tutoriels/Dumortier/cadre\\_comp.htm](http://www.ai.polymtl.ca/cohen/ELE6812/Tutoriels/Dumortier/cadre_comp.htm)
- [26] Jérôme Landré, Analyse Multirésolution Pour La Recherche Et L'indexation D'images Par Le Contenu Dans Les Bases De Données Images - Application À La Base D'images Paléontologique Trans'lyfipal
- [27] BraniVidakovic and Peter Mueller Duke University, wavelet for kids A Tutorial Introduction.
- [28] JanuszBobulski, Czestochowa University of Technology Institute of Computer and Information Science, Wavelet Transform in Face Recognition.
- [29] Sanchez Lucie Meurisse Christophe, ENSICAEN, Traitement des images par les Ondelettes.
- [30] The Discrete Wavelet Transform : [www.dtic.upf.edu/~xserra/cursos/TDP/Park-DWT.pdf](http://www.dtic.upf.edu/~xserra/cursos/TDP/Park-DWT.pdf)
- [31] Wikipedia, « Analyse en composantes principales » , [http://fr.wikipedia.org/wiki/Analyse\\_en\\_composantes\\_principales](http://fr.wikipedia.org/wiki/Analyse_en_composantes_principales)
- [32] Nicolas MORIZET, « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris », these Doctorat, Ecole Nationale Supérieure des Télécommunications, France, 2009.
- [33] M. A. Turk et A. P. Pentland, « Eigenfaces » ,2009 , [http://informatik.unibas.ch/lehre/fs09/cs331/resources/slides/eigenfaceLecture\\_handout2.pdf](http://informatik.unibas.ch/lehre/fs09/cs331/resources/slides/eigenfaceLecture_handout2.pdf)
- [34] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991,
- [35] Lindsay I Smith, « Principal Components Analysis », Ebook, [http://www.cs.otago.ac.nz/cosc453/student\\_tutorials/principal\\_components.pdf](http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf)
- [36] Pierre-Louis Gonzalez, "L'ANALYSE EN COMPOSANTES PRINCIPALES", Ebook, <http://www.cnam.fr/math/IMG/pdf/Compos-princip.pdf>
- [37] NadjetRadji, DrDalilaCherifi, Dr.H.Benterzi, "Proposed Face Recognition Method based on PCA and WPD using YCbCr Color Space", Ebook, DGEE (ex.INELEC), Université de Boumerdès, Algérie, 2009.
- [38] <http://www.enimag.co.cc/espace-partage/6-les-fpga-field-programmable-gate-array-reseau-de-portes-programmables-in-situ-expliques-par-un-electromecanicien>
- [39] <http://zone.ni.com/devzone/cda/tut/p/id/8043>
- [40] Etienne Messerli, Manuel VHDL synthèse et simulation. Version 6-a Septembre 2007.
- [41] [http://pagesperso-orange.fr/michel.hubin/physique/microp/chap\\_mp5.htm](http://pagesperso-orange.fr/michel.hubin/physique/microp/chap_mp5.htm)
- [42] Wendy S. Yambor Bruce A. Draper J. Ross Beveridge, Analyzing PCA-based Face Recognition Algorithms: Eigenvector. Selection and Distance MeasuresComputer Science DepartmentColorado State UniversityFort Collins, CO, U.S.A 80523, July 1, 2000
- [43] <http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm>
- [44] <http://www.face-rec.org/>

# Annexes

---

## A: Combining Classifiers For Face Recognition

---

# COMBINING CLASSIFIERS FOR FACE RECOGNITION

Xiaoguang Lu\*, Yunhong Wang†, Anil K. Jain\*

\*Department of Computer Science & Engineering, Michigan State University  
East Lansing, MI 48824  
{lvxiaogu, jain}@cse.msu.edu

†National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences  
P.O. Box 2728, Beijing 100080, P. R. China  
wangyh@nlpr.ia.ac.cn

## ABSTRACT

Current two-dimensional face recognition approaches can obtain a good performance only under constrained environments. However, in the real applications, face appearance changes significantly due to different illumination, pose, and expression. Face recognizers based on different representations of the input face images have different sensitivity to these variations. Therefore, a combination of different face classifiers which can integrate the complementary information should lead to improved classification accuracy. We use the sum rule and RBF-based integration strategies to combine three commonly used face classifiers based on PCA, ICA and LDA representations. Experiments conducted on a face database containing 206 subjects (2,060 face images) show that the proposed classifier combination approaches outperform individual classifiers.

## 1. INTRODUCTION

Human face recognition has a tremendous potential in a wide variety of commercial and law enforcement applications. Considerable research efforts have been devoted to the face recognition problem over the past decade [1]. Although there are a number of face recognition algorithms which work well in constrained environments, face recognition is still an open and very challenging problem in real applications.

Among face recognition algorithms, appearance-based approaches [2][3][4][5] are the most popular. These approaches utilize the pixel intensity or intensity-derived features. Several such systems have been successfully developed and installed [1][6][7][8]. However, appearance-based methods do not perform well in many real-world situations,

where the query test face appearance is significantly different from the training face data, due to variations in pose, lighting and expression. Some examples of these variations for one of the subjects in our database are illustrated in Fig. 1. While a robust classifier could be designed to handle any one of these variations, it is extremely difficult for an appearance-based approach to deal with all of these variations. Each individual classifier has different sensitivity to different changes in the facial appearance. It has been reported that each appearance-based method shows different levels of performance on different subsets of images [6], suggesting that different classifiers contribute complementary information to the classification task. A combination scheme involving different face classifiers, which integrates various information sources, is likely to improve the overall system performance.

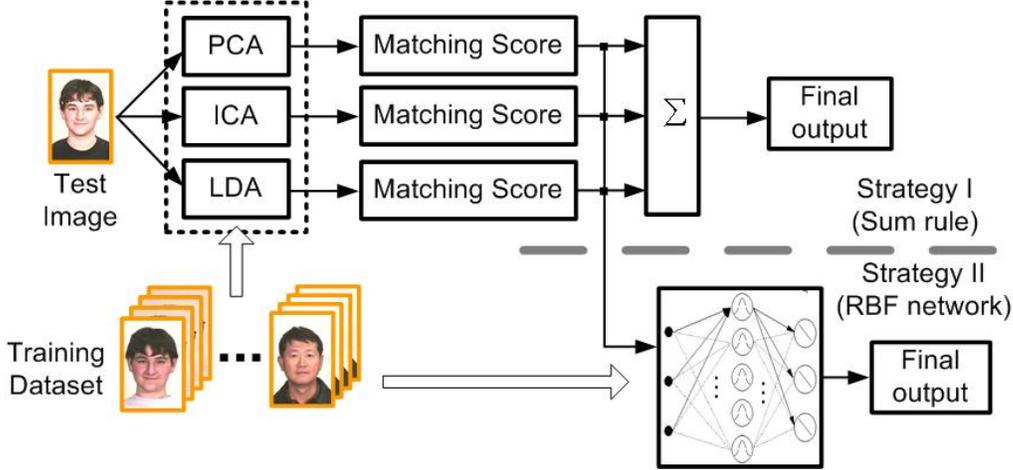


**Fig. 1.** Facial variations under different lighting conditions and facial expressions for the same subject [9].

The classifier combination can be implemented at two levels, feature level and decision level. We use the decision level combination that is more appropriate when the component classifiers use different types of features. Kittler [10] provides a theoretical framework to combine various classifiers at the decision level. Many practical applications of combining multiple classifiers have been developed. Brunelli and Falavigna [11] presented a person identification system by combining outputs from classifiers based on audio and visual cues. Jain et al. [12] integrated multiple fingerprint matchers to develop a robust fingerprint verification system. Hong and Jain [13] designed a decision fusion scheme to combine faces and fingerprint for personal identification. Marcialis and Roli [14] exploited the fusion of

---

This research was supported by NSF IUC on Biometrics (CITeR), at West Virginia University.



**Fig. 2.** Classifier combination system framework.

PCA and LDA for face verification.

We propose two combination strategies, sum rule and RBF network, to integrate the outputs of three well-known appearance-based face recognition methods, namely PCA [2], ICA [3] and LDA [4][5]. Our combination strategy is designed at the decision level, utilizing all the available information, i.e. a subset of (face) labels along with a confidence value, called the matching score provided by each of the three face recognition method.

## 2. CLASSIFIER INTEGRATION

Our combination scheme is illustrated in Fig. 2. While this framework does not limit the number of component classifiers, we currently use only three classifiers, namely, PCA, ICA and LDA. Following two strategies are provided for integrating outputs of individual classifiers, (i) the sum rule, and (ii) a RBF network as a classifier, using matching scores as the input feature vectors.

### 2.1. Appearance-based Face Classifiers

Three appearance-based classifiers, PCA [2], ICA [3] and LDA [4][5] have been implemented. In each of these approaches, the 2-dimensional face image is considered as a vector, by concatenating each row (or column) of the image. Each classifier has its own representation (basis vectors) of a high dimensional face vector space. By projecting the face vector to the basis vectors, the projection coefficients are used as the feature representation of each face image. The matching score between the test face image and training data is calculated as the cosine value of the angle between their coefficients vectors.

Let  $X = (x_1, x_2, \dots, x_i, \dots, x_N)$  represent the  $n \times N$  data matrix, where each  $x_i$  is a face vector of dimension  $n$ ,

concatenated from a  $p \times p$  face image, where  $p \times p = n$ . Here  $n$  represents the total number of pixels in the face image and  $N$  is the number of face images in the training set. The mean vector of the training images  $\mu = \sum_{i=1}^N X_i$  is subtracted from each image vector. All the three representations can be considered as a linear transformation from the original image vector to a projection feature vector, i.e.

$$Y = W^T X, \quad (1)$$

where  $Y$  is the  $d \times N$  feature vector matrix,  $d$  is the dimension of the feature vector, and  $W$  is the transformation matrix. Note that  $d \ll n$ .

1. PCA [2]. The Principal Component Analysis basis vectors are defined as the eigenvectors of the scatter matrix  $S_T$ ,

$$S_T = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T. \quad (2)$$

The transformation matrix  $W_{PCA}$  is composed of the eigenvectors corresponding to the  $d$  largest eigenvalues. After applying the projection, the input vector (face) in an  $n$ -dimensional space is reduced to a feature vector in a  $d$ -dimensional subspace.

2. ICA [3]. Bartlett et al. [3] provided two architectures based on Independent Component Analysis, statistically independent basis images and a factorial code representation, for the face recognition task. The ICA separates the high-order moments of the input in addition to the second-order moments utilized in PCA. Both the architectures lead to a similar performance. There is no special order imposed on the ICA basis vectors.

3. LDA [4][5]. The Linear Discriminant Analysis finds a transform  $W_{LDA}$ , such that

$$W_{LDA} = \arg \max_W \frac{W^T S_B W}{W^T S_W W}, \quad (3)$$

where  $S_B$  is the between-class scatter matrix and  $S_W$  is the within-class scatter matrix, defined as

$$S_B = \sum_{i=1}^c N_i (x_i - \mu)(x_i - \mu)^T, \quad (4)$$

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T. \quad (5)$$

In the above expression,  $N_i$  is the number of training samples in class  $i$ ,  $c$  is the number of distinct classes,  $\mu_i$  is the mean vector of samples belonging to class  $i$  and  $X_i$  represents the set of samples belonging to class  $i$ .

## 2.2. Integration Strategy

Kittler [10] analyzed several classifier combination rules and concluded that the sum rule (defined below) outperforms other combination schemes based on empirical observations. Unlike explicitly setting up combination rules, it is possible to design a new classifier using the outputs of individual classifiers as features to this new classifier. We adopt the RBF network [15] as this new classifier. Given  $m$  templates in the training set,  $m$  matching scores will be output for each test image from each classifier. We consider the following two integration strategies

1. Strategy I: Sum Rule. The combined matching score is calculated as

$$MS_{comb} = MS_{PCA} + MS_{ICA} + MS_{LDA}. \quad (6)$$

For a given test sample, Output the class with the largest value of  $MS_{comb}$ .

2. Strategy II: RBF network. For each test image, the  $m$  matching scores obtained from each classifier are used as a feature vector. Concatenating these feature vectors derived from three classifiers results in a feature vector of size  $3m$ . An RBF network is designed to use this new feature vector as the input to generate classification results. We adopt a 3-layer RBF network. The input layer has  $3m$  nodes and the output has  $c$  nodes, where  $c$  is the total number of classes (number of distinct faces). In the output layer, the class corresponding to the node with the maximum output is assigned to the input image. The number of nodes in the hidden layer is constructed empirically, depending on the sizes of the input and output layers.

## 3. EXPERIMENTS AND DISCUSSION

Our database is a collection of four different face databases, available in the public domain (see table 1). There are 206 subjects with 10 images per subject for a total of 2,060 images. Face images selected are near frontal and contain variations in pose, illumination and expression. Some images in the individual databases are not selected for our experiments; these face images have out-of-plane rotation by more than 45 degrees in the NLPR+MSU database and face images with occlusions due to sun glasses or a scarf in the AR database. Sample images from the databases are shown in Fig. 3. Face images are closely cropped to include only the internal facial structures such as the eyebrows, eyes, nose and mouth, and aligned by the centers of the two eyes. All cropped images are resized to  $42 \times 42$ . Each image vector is normalized to be of unit length.

**Table 1.** Database description.

Face database	No. of subjects	Variations included
ORL [16]	40	Slight pose and expression
Yale [9]	15	Illumination and expression
AR [17]	120	Illumination and expression
NLPR+MSU (collected by the authors)	31	Slight pose and expression



**Fig. 3.** Representative face images in the database. (a) ORL, (b) Yale, (c) AR and (d) NLPR+MSU.

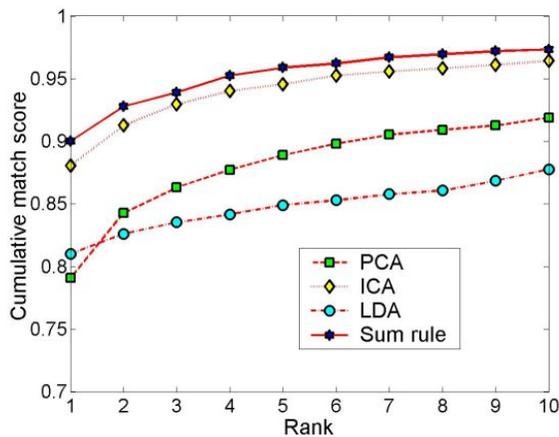
The entire face database is divided into two parts. Nine images of each subject are used to construct the training data and the remaining one is used for testing. This partition is repeated 10 different times so that every image of the subject can be used for testing. The classification accuracy is the average of these ten different tests.

All the individual classifiers use the cosine value of the angle between the two projection coefficient vectors (one from the database image and the other from the test image) as the matching score. Database image with the best match is used to determine the classification of the input image. The sum rule is applied to the matching score outputs of the three classifiers. The database image with the maximum

sum score is output as the final result. The recognition accuracies of different face recognition approaches are listed in table 2. The cumulative match score vs. rank curve [6] is used to show the performance of each classifier, see Fig. 4. Since our RBF network outputs the final label, no rank information is available. As a result, we cannot compute the cumulative match score vs. rank curve for RBF combination.

**Table 2.** Recognition accuracy of different classifiers.

PCA	ICA	LDA	Sum rule	RBF based
79.1%	88.1%	81.0%	90.0%	90.2%



**Fig. 4.** Cumulative match score vs. rank curve for the sum rule.

Table 2 and figure 4 show that the combined classifiers, based on both the sum-rule and RBF network, outperform each individual classifier.

#### 4. CONCLUSIONS AND FUTURE WORK

An integration scheme, which combines the output matching scores of three well-known face recognition approaches, is proposed to improve the performance of a face identification system. Two combination strategies, sum rule and RBF-based integration, are implemented to combine the output information of three individual classifiers, namely PCA, ICA and LDA. The proposed system framework is scalable; other face recognition modules can be easily added into this framework. Experimental results are encouraging, illustrating that both the combination strategies lead to more accurate face recognition than that made by any one of the individual classifiers. We are currently investigating the weighted sum rule based on the user-specific matching score distribution.

#### 5. REFERENCES

- [1] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face recognition: A literature survey," *CVL Technical Report, University of Maryland*, Oct. 2000, <ftp://ftp.cfar.umd.edu/TRs/CVL-Reports-2000/TR4167-zhao.ps.gz>.
- [2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Mar. 1991.
- [3] M.S. Bartlett, H.M. Lades, and T.J. Sejnowski, "Independent component representations for face recognition," in *Proceedings of the SPIE*, 1998, vol. 3299, pp. 528–539.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
- [5] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.
- [6] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [7] Identix, <<http://www.identix.com/>>, Minnetonka, MN.
- [8] Viisage, <<http://www.viisage.com/>>, Littleton, MA.
- [9] Yale University face database, <<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>>.
- [10] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [11] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 10, pp. 955–966, Oct. 1995.
- [12] A. K. Jain, S. Prabhakar, and S. Chen, "Combining multiple matchers for a high security fingerprint verification system," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1371–1379, 1999.
- [13] L. Hong and A.K Jain, "Integrating faces and fingerprint for personal identification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1295–1307, 1998.
- [14] G. L. Marcialis and F. Roli, "Fusion of lda and pca for face verification," in *Biometric Authentication*, Jun. 2002, vol. LNCS 2359, pp. 30–37.
- [15] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, UK, 1995.
- [16] Ferdinando Samaria and Andy Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, Dec. 1994.
- [17] A.M. Martinez and R. Benavente, "The ar face database," *CVC Tech. Report # 24*, Jun. 1998.