

République Algérienne Démocratique et Populaire

Ministère de L'enseignement Supérieur et de la Recherche Scientifique



Université SAAD DAHLEB de BLIDA
Faculté des sciences
Département Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de Master

Spécialité:

Informatique

Option :

Ingénierie du logiciel

Par : HANNACHI Lilia

Thème

***Compression /Décompression des empreintes
digitales selon le format FBI-Wavelet Scalar
Quantization***

Proposé par :

Dr. Abdelhak Harzallah.

Mr. Abdelhalim Khababa.

Suivi par :

Dr. Benblidia .Nadjia

Président: Mme. Requinig

Examinateur : Mme. Benseltiti

Année universitaire 2008-2009.

MA-004-03-1

Remerciements

*Je tiens tout particulièrement à remercier ma promotrice M^{elle}
N.BENBLIDA et mes encadreurs M^r KHABABA et
M^r HARZALLAH pour ses conseils et ses orientations.*

*Mes remerciements à Mr ATLA, M^r SAIFI et Mr MOSTFAI pour
toute l'aide qu'ils ont apporté.*

*J'exprime mon gratitude à tous ce qui ont contribué de près ou de
loin à l'élaboration de ce travail .*

*Enfin je remercie tous les membres du jury d'avoit accepter de
siège à la soutenance de ce mémoire*

Table des matières



Introduction	1
<i>CHAPITRE I : LA BIOMETRIE ET LES EMPREINTES DIGITALES</i>	
I.1. La biométrie.....	4
I.1.1. Qu'est ce que la biométrie?	4
I.1.2. Les différentes techniques biométriques.....	5
I.1.3. Les empreintes digitales.....	7
I.1.3.1. Historique	7
I.1.3.2. Caractéristiques des empreintes.....	7
I.2. Structure d'un système complet de reconnaissance d'empreintes	9
I.2.1. Principe général.....	9
I.2.2. L'acquisition de l'empreinte.....	10
I.2.3. Le traitement de l'image et l'extraction de la signature.....	11
I.2.4. Le stockage et la phase d'appariement	11
I.3. Les problèmes posés par l'utilisation de la biométrie	12
I.4. Conclusion.....	13
<i>CHAPITRE II : LES DIFFERENTES TECHNIQUES DE COMPRESSION</i>	
Introduction	14
II.1 Généralités sur la compression d'images	14
II.1.1 Taux de compression et redondance.....	14
II.1.2 Critères psychovisuels et compression.....	15
II.1.2.1 Contours et Texture.....	15
II.2 Codage sans perte.....	15
II.2.1 Codage d'Huffman	16
II.2.2 Codage arithmétique.....	16
II.2.3 L'algorithme LZW	17
II.2.4 Codage par plage	18
II.2.5 Codage par prédiction linéaire	18

II.3 Codage avec pertes.....	18
II.3.1 Quantification.....	19
II.3.1.1 Scalaire.....	19
II.3.1.2 Vectorielle.....	19
II.3.2 Codage prédictif avec pertes.....	20
II.3.3 Codage par transformation.....	20
II.4 Domaine spatial.....	20
II.4.1 Quadtree.....	20
II.4.2 Décomposition en plans binaire.....	21
II.4.3 Fractales.....	22
II.5 Domaine fréquentiel.....	22
II.5.1 Les standards.....	23
II.5.1.1 JPEG.....	23
II.5.1.2 JPEG-LS.....	25
II.5.1.3 JPEG2000.....	26
II.6 Conclusion.....	27

*CHAPITRE III : COMPRESSION SELON LE FORMAT FBI-WAVELET SCALAR
QUANTIZATION*

Introduction.....	28
III.1 Les Ondelettes.....	28
III.1.1 Fréquence locale.....	28
III.1.2 Représentations temps–fréquence et temps–échelle...	30
III.1.3 Transformée en ondelettes continue.....	32
III.1.4 La transformée en ondelettes discrète.....	33
III.1.5 Analyses multi-résolutions.....	34
III.1.5.1 Cadre théorique.....	34
III.1.5.2 Les base d’ondelettes.....	35
III.1.5.3 Transformée en ondelettes.....	36
III.1.5.4 Filtres duaux, ondelettes duales.....	37
III.1.5.5 L’algorithme de la transformée en ondelettes rapide.	38
III.1.5.6 Les ondelettes orthogonales.....	39
III.1.6 Décroissance des coefficients, régularité et approximation ...	40
III.2 Le format FBI – Wavelet Scalar Quantization (WSQ).....	42

III.2.1 Un aperçu général sur l'algorithme WSQ	42
III.2.1.1 La transformée en ondelettes	44
III.2.1.2 La quantification	47
III.2.1.3 Le codeur entropique	48
III.2.2 Comparaison entre WSQ et JPEG 2000	48
III.3 Conclusion.....	50
<i>CHAPITRE IV : ANALYSE, CONCEPTION ET REALISATION</i>	
IV.1 Etude et présentation du " <i>First-Compression</i> "	51
IV.1.1 Introduction.....	51
IV.1.2 Présentation de <i>First-Compression</i>	51
IV.1.3 Description de la structure de <i>First-Compression</i>	52
IV.1.3.1 Les classes du Codeur	52
IV.1.3.2 Les classes du Décodeur.....	67
IV.2 Tests en <i>First-Compression</i>	74
IV.2.1 Introduction.....	74
IV.2.2 Le taux de compression	74
IV.2.3 Les mesures de fidélité	74
IV.3 Présentation de l'application.....	76
IV.3.1 Quelques interfaces réalisées.....	76
CONCLUSION GENERALE	93
BIBLIOGRAPHIE.....	94

Depuis que le FBI a commencé en 1924 à garder des enregistrements d'empreintes digitales, sa collection est passée d'un nombre initial de 810.000 cartes jusqu'à atteindre à présent plus de 25 millions de cartes [1]. L'archivage de cette information sous forme de cartes imprimées a des inconvénients évidents quand il s'agit de transmission, de stockage, et d'analyse automatisée des empreintes digitales.

Néanmoins, il y a de nombreux avantages de stocker et transmettre des dossiers d'empreintes digitales dans un certain type de format numérique, et bon nombre d'administrations municipales et étatiques ont mis en œuvre différents systèmes commerciaux d'imagerie numérique pour enregistrer les données des empreintes digitales. Cela a conduit à des problèmes de compatibilité du fait de l'utilisation concurrentielle par plusieurs propriétaires de différents matériaux et de formats de données. Un autre facteur majeur de l'intérêt porté par le FBI à la numérisation des empreintes digitales, est l'amélioration du délai de leur réponse aux requêtes du système judiciaire. Cela nécessitera à la fois une transmission rapide des dossiers d'arrêt, tels que les empreintes digitales, et l'automatisation de la vérification des antécédents; une tâche qui sera facilitée par un système intégré « Automated Fingerprint Identification » actuellement en cours de développement.

En réponse à ces questions, les services d'information de la division criminelle du FBI (CJIS) ont développé des normes pour la numérisation des empreintes digitales, en coopération avec le fournisseur commercial et la justice pénale, ainsi qu'avec l'aide de l'Institut National des Normes et des Technologies (NIST) et du Laboratoire National de Los Alamos (LANL) [9].

Les images des empreintes digitales sont numérisées avec une résolution de 500 pixels / pouce et 256 niveaux de gris (8 bits) ; les niveaux de gris des images paraissent plus naturels à la vue de l'homme, ce qui n'est pas le cas des images des « bit-mapped », et entraînent un meilleur discernement subjectif par les examinateurs des empreintes digitales [4]. Le stockage de données et la transmission des exigences imposées par ce niveau de résolution sont considérables: une parcelle d'empreinte digitale de 1,5 pouce seule est transformée en près de 600 kilo-octets d'informations numériques ; une carte entière (l'impression des 10 doigts simultanément) produit près de 10 méga-octets de données [3]. À ce rythme, la numérisation des fonds documentaires du FBI en cours aboutirait à quelques 250 téraoctets de données d'archives [1]. Il ya

également d'importantes considérations de transmission: la transmission électronique d'une seule carte de 10 Mo à travers un modem à un taux de 9600 bits / seconde prendrait près de 3 heures [10]. Toutefois, tout cela est encore plus rapide que les services de livraison de nuit et élimine tout danger de perte ou d'endommagement de carte par la poste et qui serait irremplaçable. L'ampleur de la base de données d'empreintes digitales est donc telle que le FBI a fait de la compression de données une partie de la norme de la numérisation. L'algorithme de compression adaptative choisi par CJIS est fondé sur la quantification scalaire uniforme, sur une sous bande de l'image de la transformation en ondelettes et est dénommé wavelet scalar quantization (WSQ) standard [4].

L'analyse par ondelettes a été introduite au début des années 1980 dans le contexte de l'analyse des signaux et de l'exploration pétrolière [34]. Il s'agit à l'époque de donner une représentation des signaux permettant la mise en valeur simultanément des informations temporelles et fréquentielles (localisation temps-fréquence).

L'idée originale sur laquelle sont basées les ondelettes est apparue vers les années 1940 grâce au physicien Denis Gabor, (prix Nobel de physique, en 1971, pour l'invention de l'holographie), qui a introduit la notion de la transformée de Fourier à fenêtre glissante dans le but de remédier au problème de localisation temps-fréquence en proposant de multiplier le signal par une fonction localisée dans le temps (fenêtre) et ensuite appliquer la transformée de Fourier [39].

L'inconvénient de cette transformée est que la taille et la forme de la fenêtre sont inchangées au cours de l'analyse. Or, pour étudier un signal qui a, en général, une allure irrégulière, il est potentiellement intéressant de pouvoir changer la forme de la fenêtre analysante en temps. C'est à partir de là que sont nées les ondelettes qui s'adaptent d'elles mêmes à la taille et aux caractéristiques qu'elles recherchent.

Dés lors, les ondelettes qui ont été créés pour résoudre des problèmes posés par la sismique ne cessent de se développer, tant du point de vue pratique que du point de vue théorique, par des personnes telles Y. Meyer , I. Daubechies , S. Mallat et autres pour ouvrir ensuite les ondelettes sur de nombreux champs d'applications autres que la sismologie ou les mathématiques. Nous citons par exemple l'apport des ondelettes en compression d'images imagerie médicale, turbulence, la téléphonie vidéo, les systèmes radar, le stockage numérique des empreintes digitales (effectué par le FBI) et également le stockage ou la transmission de résultats de mesures sismiques, ...etc[40].

L'Algérie est en train de réaliser une base dans le domaine de la sécurité où il y a un manque. Notre travail, qui porte sur la compression des empreintes digitales et des images faciales, fait partie d'un grand projet. Celui-ci se décompose en différentes parties telles que la compression, l'extraction des caractéristiques, la comparaison et l'identification. Il s'agit d'étudier et de concevoir un système de compression d'empreintes en utilisant l'analyse multirésolution et la quantification scalaire. Pour réaliser notre travail, nous avons alors choisi d'utiliser :

- le langage UML pour ses performances de conception et de modélisation et pour sa facilité de compréhension,
- Le langage c++ qui est proche de divers langages et qui facilite par conséquent la conversion,
- Le logiciel c++ Borland car il offre une implémentation et une visualisation de bonne qualité.

Notre mémoire se compose de quatre chapitres présentés comme suit :

Le *chapitre 1* intitulé *la biométrie et les empreintes digitales*, dans lequel nous commencerons par présenter la biométrie de manière générale ainsi que les diverses applications qui en découlent, en insistant plus particulièrement sur l'utilisation des empreintes digitales. Puis nous détaillerons les différentes étapes composant un système complet de reconnaissance d'empreintes.

Le *chapitre 2* intitulé *les différentes techniques de compression*, dans lequel nous présenterons les différents critères pour comprimer une image, ainsi que la classification des méthodes de codage. Nous terminerons par exposer les standards de compression d'images fixes les plus répandus : JPEG basé sur la DCT, JPEG-LS, JPEG2000 basé sur la DWT.

Le *chapitre 3* intitulé *la compression selon le format « FBI-Wavelet Scalar Quantization »*, dans lequel nous commencerons par définir les différentes notions de la transformée en ondelettes avant d'entrer dans le vif du sujet qui consistera en la présentation de la WSQ (Wavelet Scalar Quantization).

Le *chapitre 4* intitulé *Conception, modélisation et réalisation*, dans lequel sera présentée la conception de notre système de compression. Nous modéliserons les différentes fonctionnalités à atteindre dans notre application et nous présenterons le travail réalisé sous forme de figures et fenêtres donnant une idée globale sur notre application et son utilisation.

Pour conclure, nous verrons quels sont les buts atteints et ce qu'il reste à réaliser.

I.1. La biométrie

I.1.1. Qu'est ce que la biométrie?

Depuis quelques décennies l'explosion de l'informatique et des réseaux de communication a fait augmenter de manière significative le besoin d'identification des personnes. Jusqu'à présent les méthodes usuelles d'identification sont basées sur ce que l'on possède (carte d'identité, carte à puce, badge magnétique) ou sur ce que l'on sait (mot de passe, code PIN) mais ces méthodes posent de gros problèmes de fiabilité (falsification de document, oubli de son code, décryptage du mot de passe via des logiciels spécifiques). Depuis les récents actes terroristes et les menaces qui pèsent sur de nombreux pays, une identification fiable des personnes est devenue un problème majeur pour des raisons de sécurité (contrôle aux frontières, accès aux lieux publics, transport). Tous ces problèmes ont ainsi provoqué un développement accru des techniques biométriques d'identification [1], [2] comme en témoigne l'étude des perspectives du marché de la biométrie (Figure I-1).

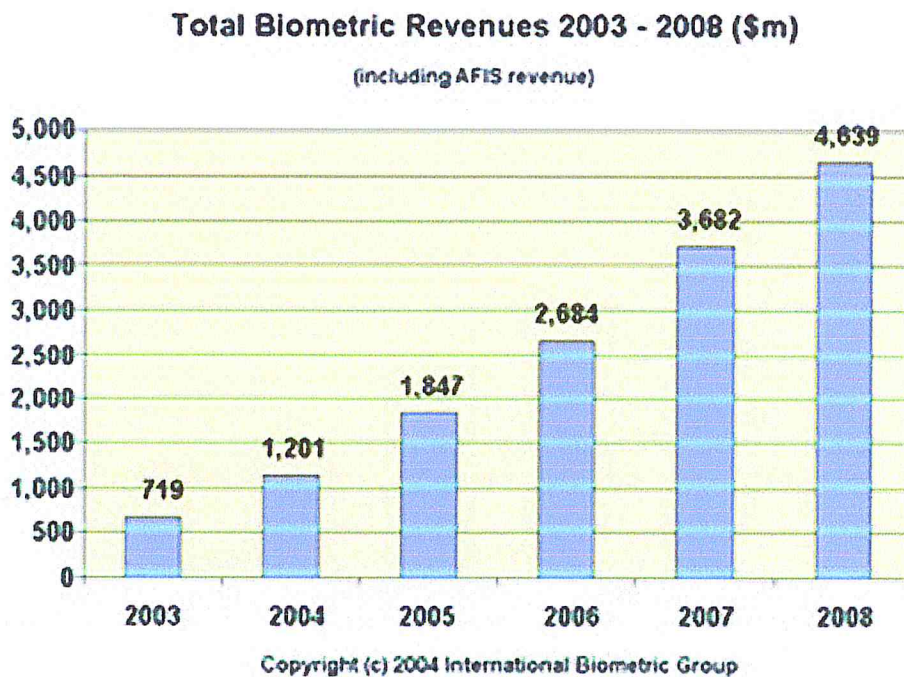


Figure I-1: Evolution du marché international de la biométrie [3]

La biométrie consiste en l'analyse mathématique des caractéristiques biologiques d'une personne et a pour objectif de déterminer son identité de manière irréfutable. Contrairement à *ce que l'on sait* ou *ce que l'on possède*, la biométrie est basée sur *ce que l'on est* et permet ainsi d'éviter la duplication, le vol, l'oubli ou la perte. Les caractéristiques utilisées doivent être *universelles* (c'est-à-dire communes à tous les individus), *uniques* (pour pouvoir différencier deux individus) et *permanentes* (c'est-à-dire invariantes dans le temps pour chaque individu).

Il existe deux types de systèmes de reconnaissance biométrique: ceux basés sur la *vérification* et ceux basés sur l'*identification*. La vérification, également appelée authentification, consiste à confirmer ou infirmer l'identité d'une personne (suis-je celui que je

prétends être?). Il s'agit d'une comparaison du type "un contre un"; les caractéristiques de l'individu sont comparées à celles présentes dans un enregistrement de référence. Quant à l'identification elle permet d'établir l'identité d'une personne (qui suis-je?) à partir d'une base de données, il s'agit d'une comparaison du type "un contre plusieurs".

L'application qui vient immédiatement à l'esprit est l'utilisation des empreintes digitales par les forces de l'ordre pour fichier les criminels. A titre d'exemple le FBI [4] possède une base de données de 250 millions d'empreintes.

Outre l'utilisation policière, la biométrie apporte des solutions dans de nombreuses applications telles que le contrôle d'accès à un lieu physique (aéroport, gare, casino), la surveillance (systèmes de vidéosurveillance des rucs, contrôle des détenus dans les prisons), le contrôle d'accès à un espace virtuel (ex. accès à un réseau local pour éviter la fraude par mot de passe), le paiement en ligne (pour sécuriser la transaction) .

I.1.2. Les différentes techniques biométriques

Parmi les différentes techniques biométriques existantes, on distingue trois catégories:

- L'analyse morphologique: les empreintes digitales, l'iris de l'œil, la forme de la main, les traits du visage, le réseau veineux de la rétine
- L'analyse des traces biologiques: l'ADN, le sang, la salive, l'urine, l'odeur, la thermographie
- L'analyse comportementale: la reconnaissance vocale, la dynamique de frappe au clavier, la dynamique de signature, la manière de marcher ...

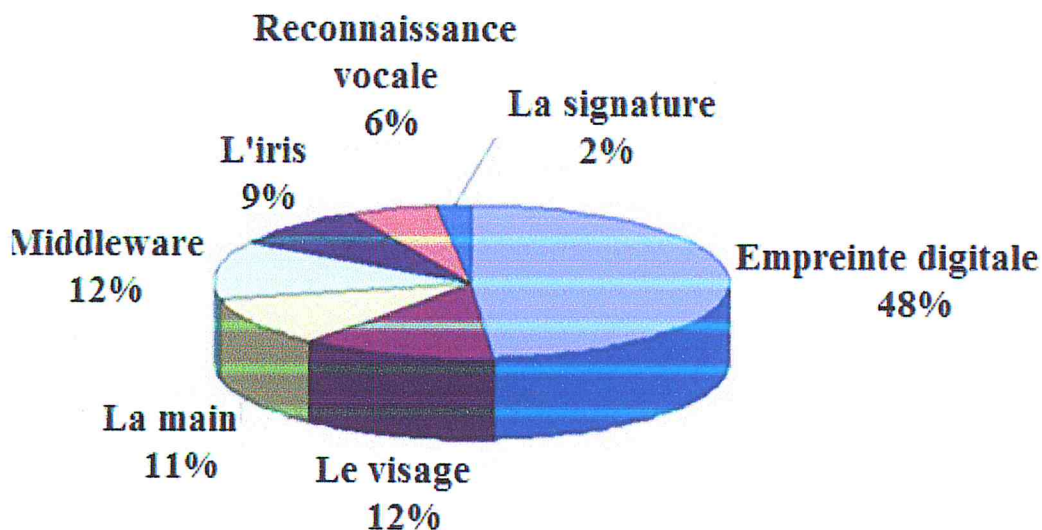


Figure I-2: Part de marché des différentes méthodes biométriques [3]

Parmi les nombreuses méthodes d'identification biométrique, l'utilisation des empreintes digitales est la méthode la plus aboutie avec 48% de parts de marché; nous y reviendrons plus en détail dans la section suivante. Néanmoins d'autres méthodes commencent à trouver leur place sur le marché de la biométrie (voir Figure I-2 et Figure I-3):

- La forme du visage [5], [6] arrive en deuxième position avec 12% de parts de marché. Plusieurs parties du visage (joues, yeux, nez, bouche) sont extraites d'une photo ou

d'une vidéo et analysées géométriquement (distances entre différents points, positions, formes). Le problème de cette méthode vient des possibles perturbations pouvant transformer le visage (maquillage, faible luminosité, présence d'une barbe ou de lunettes, expression faciale inhabituelle, changement avec l'âge).

- La *géométrie de la main* [7]: jusqu'à 90 caractéristiques de la main sont mesurées (forme de la main et des articulations, longueur et largeur des doigts, longueurs inter articulations). Le taux d'erreurs dans la reconnaissance est assez élevé, en particulier pour des personnes appartenant à une même famille en raison d'une forte ressemblance. De plus, la forme de la main évolue beaucoup avec l'âge.
- L'*iris* [8], [9] est une technique extrêmement fiable car l'iris contient une infinité de points caractéristiques (ensemble fractal), la fraude étant néanmoins possible en utilisant des lentilles.

L'acquisition de l'iris est effectuée au moyen d'une caméra pour pallier aux mouvements inévitables de la pupille. Elle est très sensible (précision, reflet) et relativement désagréable pour l'utilisateur car il doit rester grand ouvert et il est éclairé par une source lumineuse pour assurer un contraste correct.

- La *reconnaissance vocale* [10], [11]: les caractéristiques du timbre de la voix et de la prononciation sont analysées. La qualité de l'enregistrement peut poser problème et il est possible de frauder avec un échantillon vocal préenregistré.
- La *dynamique du tracé de la signature* [12]. Il s'agit d'une analyse comportementale où différents éléments (mesure de la vitesse, ordre d'écriture, pression exercée, accélérations) sont mesurés lors de la signature. La falsification est possible en passant par une phase d'apprentissage, la signature peut varier selon le stress de l'utilisateur.

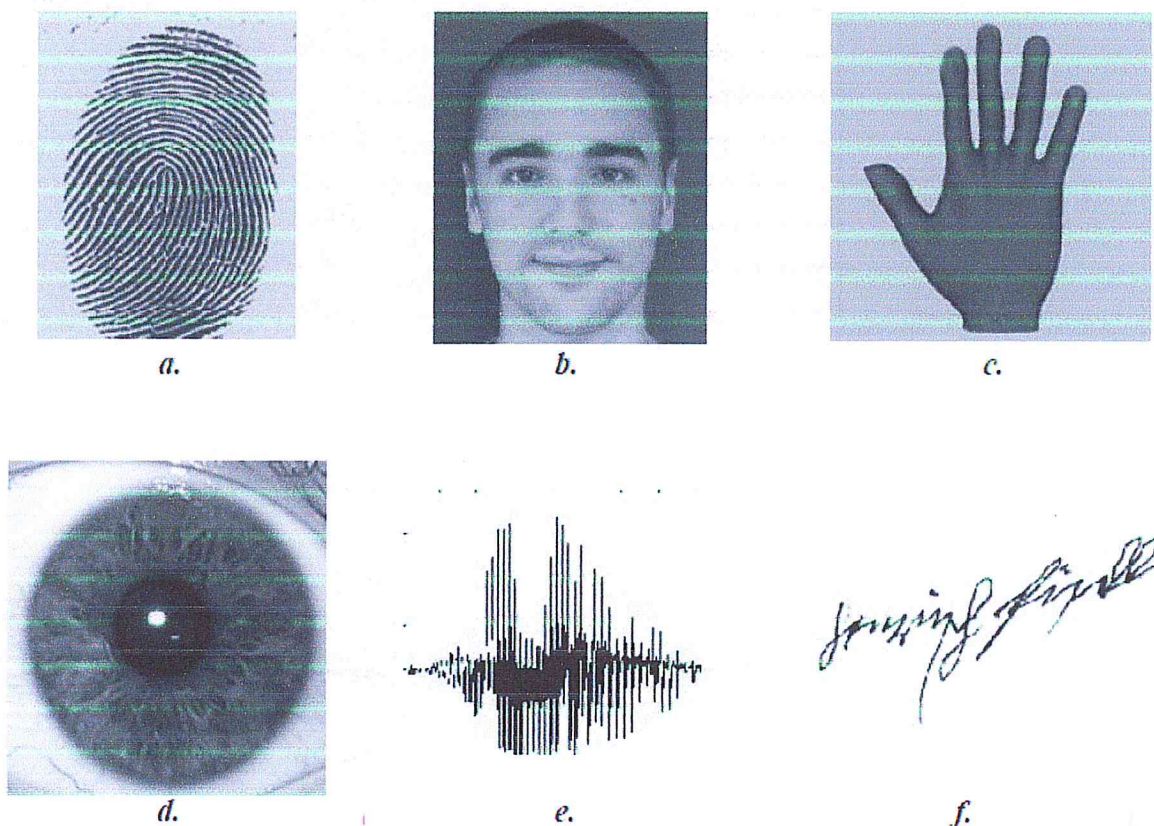


Figure I-3: Exemples de différentes caractéristiques biométriques : empreinte digitale (a), visage (b), main (c), iris (d), empreinte vocale (e), signature (f)

Certaines de ces techniques sont très prometteuses (iris) et commencent juste à émerger, d'autres sont encore au stade expérimental (analyse comportementale). Mais l'utilisation des empreintes digitales reste la méthode la plus aboutie actuellement.

I.1.3. Les empreintes digitales

I.1.3.1. Historique

Les premières traces d'utilisation d'empreintes digitales ont été découvertes en Egypte et datent de l'époque des pyramides il y a plus de 4000 ans [6]. Les Chinois ont aussi utilisé très tôt ce moyen pour signer les documents officiels (le plus vieux document signé date du troisième siècle avant Jésus Christ) mais ils ne savaient sûrement pas que les empreintes étaient uniques pour chaque personne et permettaient ainsi une identification fiable [19]. C'est en 1856 que l'anglais William Hershel, après avoir utilisé les empreintes en guise de signature sur la population indienne qu'il dirigeait, commença à comprendre que les empreintes étaient uniques et constantes dans le temps [6]. En 1888 le britannique Francis Galton publia une étude sur les empreintes digitales où il établit leurs caractéristiques (unicité, invariance, minuties, classification) et en 1901 la technique d'identification au moyen des empreintes fut adoptée officiellement en Angleterre dans le système judiciaire [11]. Cette technique fut ensuite largement développée dans les enquêtes criminelles et permit de résoudre un bon nombre d'affaires. De nos jours les empreintes sont toujours largement utilisées et reconnues comme méthode d'identification fiable.

I.1.3.2. Caractéristiques des empreintes

Une empreinte digitale est constituée d'un ensemble de lignes localement parallèles formant un motif unique pour chaque individu (Figure I-4), on distingue les *stries* (ce sont les lignes en contact avec une surface au toucher) et les *sillons* (ce sont les creux entre deux stries). Les stries contiennent en leur centre un ensemble de *pores* régulièrement espacés.

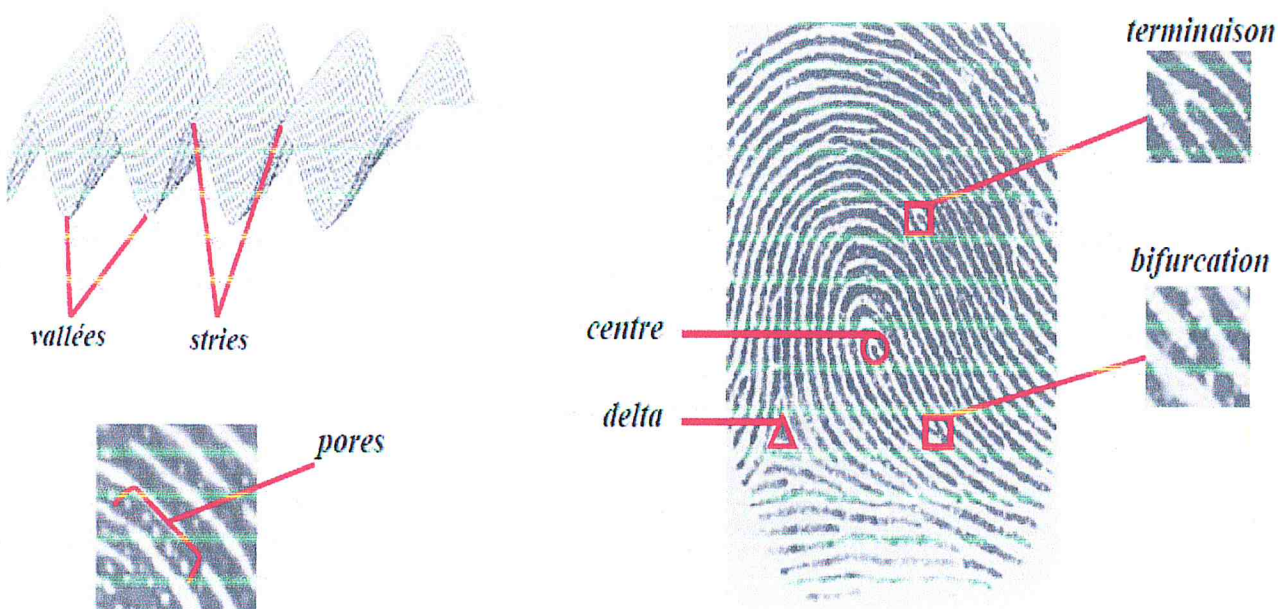
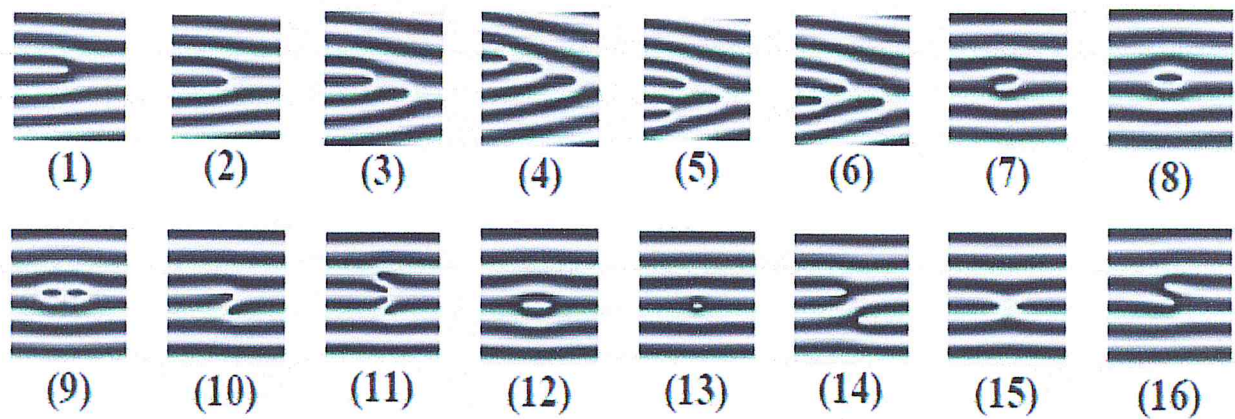


Figure I-4: Caractéristiques d'une empreinte digitale.

Chaque empreinte possède un ensemble de points singuliers globaux (les *centres* et les *deltas*) et locaux (les *minuties*). Les centres correspondent à des lieux de convergence des stries tandis que les deltas correspondent à des lieux de divergence. Une étude (Figure I-5) a montré l'existence de seize types de minuties différentes mais en général les algorithmes ne s'intéressent qu'aux bifurcations et terminaisons qui permettent d'obtenir les autres types par combinaison.



1.	terminaison	9.	boucle double
2.	bifurcation simple	10.	pont simple
3.	bifurcation double	11.	pont jumeau
4.	bifurcation triple I	12.	intervalle
5.	bifurcation triple II	13.	point isolé
6.	bifurcation triple III	14.	traversée
7.	crochet	15.	croisement
8.	boucle simple	16.	tête bêche

Figure I-5: Les différents types de minutie.

La position et le nombre de centres et de deltas permettent de classifier les empreintes en catégorie selon leur motif général. On distingue principalement trois grandes familles (voir Figure I-6) :

- Les boucles (*loop*) représentent 65% des empreintes rencontrées.
- Les spires (*whorl*) représentent 30% des empreintes rencontrées.
- Les arches (*arch*) représentent 5% des empreintes rencontrées.

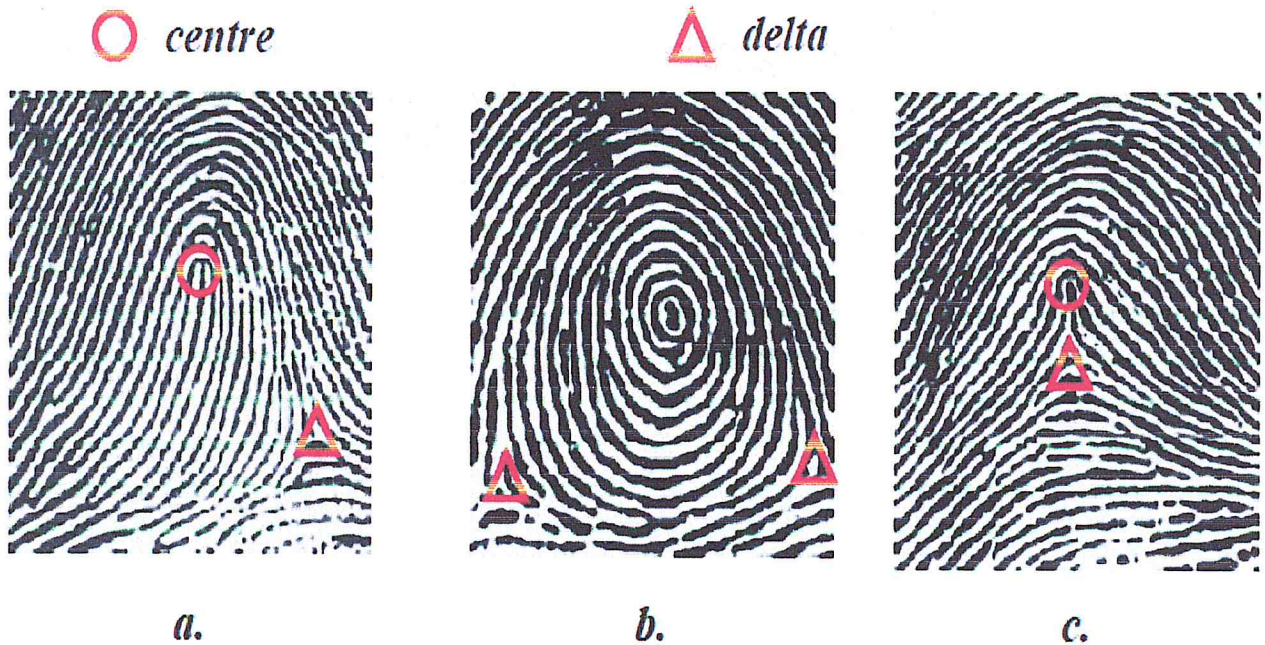


Figure I-6: Les trois principales classes d'empreinte, boucle (a), spire (b), arche (c)

L'ensemble formé par la disposition des points singuliers constitue un motif unique pour chaque individu ; en effet il a été montré [13] que l'empreinte digitale se forme au cours du troisième mois de la vie fœtale, le motif général est influencé par les gènes héréditaires mais l'apparition des détails (minuties) est créée de manière accidentelle par des pressions variables aléatoires sur les surfaces tactiles. Ainsi l'empreinte est unique pour tout individu, y compris pour des vrais jumeaux et il a été montré que les méthodes de reconnaissance actuelles permettent d'identifier efficacement les jumeaux [14]. De plus les empreintes une fois formées ne changent plus au cours de la vie d'une personne, ces deux caractéristiques en font un moyen de reconnaissance très efficace.

I.2. Structure d'un système complet de reconnaissance d'empreintes

I.2.1. Principe général

Un système automatique complet de reconnaissance d'empreintes digitales est une chaîne de processus qui à partir du doigt d'un utilisateur en entrée renvoie un résultat en sortie, permettant ainsi à l'utilisateur d'accéder ou non à des éléments nécessitant une protection. La réalisation d'un tel système a fait l'objet de très nombreuses recherches et des méthodes très différentes de traitement ont été proposées [16]. Néanmoins ces systèmes répondent toujours à la même structure (Figure I-7).

La première phase permet d'obtenir une image de l'empreinte de l'utilisateur (*acquisition*), laquelle va subir un prétraitement pour extraire l'information utile de l'image (*signature*) suivi éventuellement d'un traitement supplémentaire permettant d'éliminer de possibles fausses informations qui se seraient glissées entre temps dans la chaîne de traitement. Ensuite si l'utilisation du système consiste juste à créer une base de données (*stockage*) la signature est éventuellement compressée puis stockée dans la base de données au moyen d'une technique d'archivage (*classification*).

Pour un système d'identification l'ensemble des empreintes présentes dans la base de données pouvant correspondre à celle de l'utilisateur (modèle identique) sont désarchivées et comparées (*appariement*) une à une avec celle de l'utilisateur. Si une éventuelle

correspondance est trouvée, des informations personnelles concernant l'utilisateur sont renvoyées par le système. Dans le cas d'un système de vérification il n'y a qu'une seule comparaison et un résultat binaire est renvoyé, permettant l'acceptation ou le rejet de l'utilisateur.

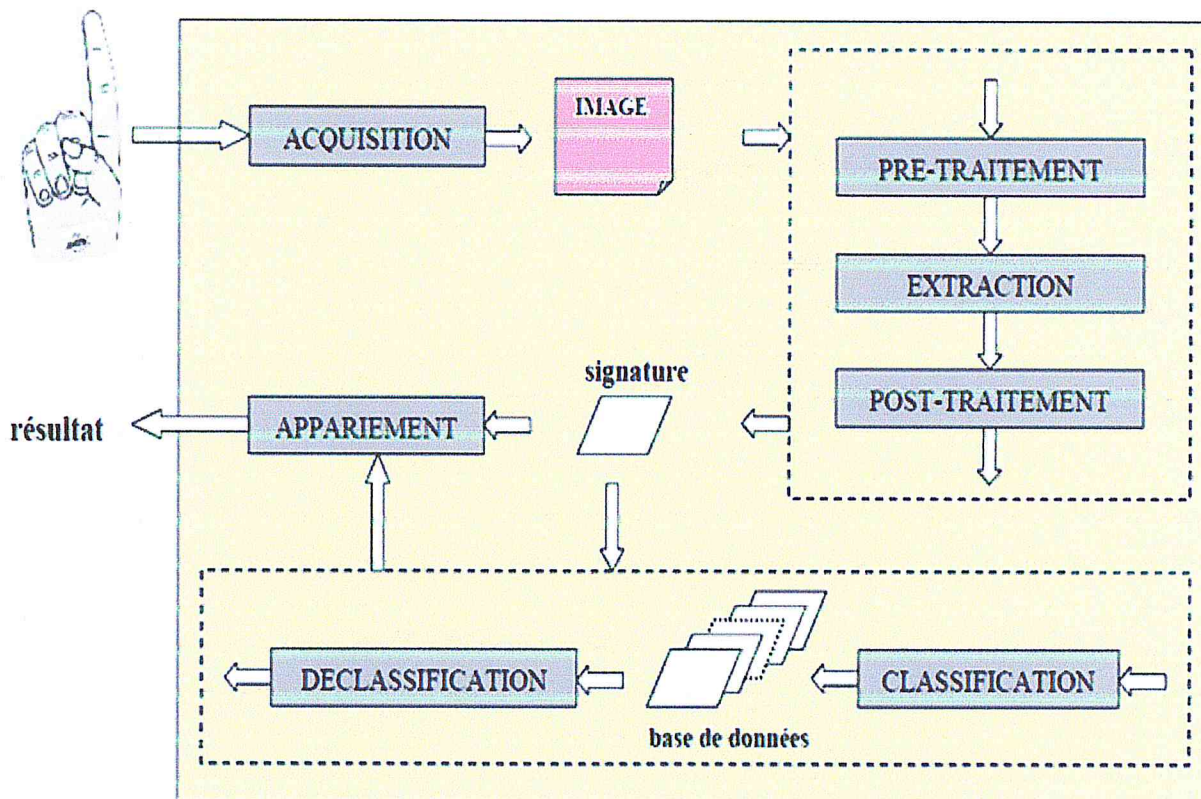


Figure I-7: Architecture générale d'un système complet de reconnaissance d'empreintes

I.2.2. L'acquisition de l'empreinte

La première phase d'un système de reconnaissance consiste à obtenir une image de l'empreinte du doigt. Longtemps le seul moyen existant a été l'utilisation du papier et de l'encre ce qui a rendu la tâche de reconnaissance très lourde. En effet, la qualité de l'image était plutôt mauvaise (plusieurs acquisitions étaient nécessaires) et l'extraction de la signature était effectuée visuellement par un expert (processus très long et pénible). Heureusement avec le développement de l'informatique et de la microélectronique de nouveaux moyens d'acquisition ont fait leur apparition, permettant ainsi d'accélérer la chaîne de traitement en l'automatisant (un capteur dédié fournit directement une image numérique).

Dans le cas d'enquêtes criminelles, les empreintes sont majoritairement utilisées. En effet un doigt contaminé par une substance telle que du sang, de l'encre ou de la graisse va laisser une trace au contact d'un support solide. Un doigt propre va également laisser des traces provoquées par les sécrétions naturelles des glandes de la peau. Dans un lieu où s'est produit un crime, les enquêteurs privilégient donc la recherche d'empreintes éventuellement laissées par le malfaiteur. Ce type d'empreintes est dénommé *empreinte latente*. Elles sont détectées au moyen d'un procédé chimique sophistiqué puis elles sont photographiées à haute résolution pour permettre l'automatisation de l'extraction de la signature. La très mauvaise qualité de ce genre d'empreintes peut nécessiter la confirmation visuelle d'un expert.

I.2.3. Le traitement de l'image et l'extraction de la signature

Lors de l'acquisition de l'empreinte, l'image obtenue contient souvent des parasites ayant des origines diverses:

- Les substances parasites présentes sur le doigt (encre, graisse, saletés).
- La personne (cicatrices, métiers manuels, âge).
- L'environnement où se produit l'acquisition (température de l'air, degré d'humidité).
- Les caractéristiques spécifiques du moyen d'acquisition utilisé.

Pour permettre une reconnaissance fiable, un prétraitement est alors nécessaire pour améliorer la qualité de l'image obtenue et éviter les erreurs. L'image est donc filtrée et les caractéristiques locales des stries (direction et fréquence) sont généralement utilisées pour augmenter l'efficacité du traitement [15].

La reconnaissance d'empreintes est basée sur l'extraction de la *signature*. La signature d'une empreinte digitale correspond à l'information utile nécessaire à l'identification fiable de la personne. La très grande majorité des techniques de reconnaissance sont basées sur la détection locale des minuties [16] et l'extraction de leurs caractéristiques (type, direction locale,) car historiquement, c'est la technique qui a toujours été utilisée par les experts humains. Certains algorithmes permettent d'extraire l'information des minuties directement à partir de l'image en niveaux de gris en suivant le maximum local des stries [17] néanmoins cela nécessite une bonne qualité d'image à la base, c'est pourquoi la plupart des algorithmes préfèrent travailler sur un squelette binaire de l'image où l'extraction est grandement facilitée. L'inconvénient de cette technique est de produire la détection d'un nombre important de fausses minuties, un post-traitement est alors nécessaire pour les éliminer [18]. Il arrive également que les pores de l'empreinte [19] soient utilisés, mais cela nécessite des images de très hautes définitions et reste donc peu utilisé.

La structure globale de l'empreinte peut aussi être utilisée [20], mais les résultats sont généralement moins précis qu'avec les caractéristiques locales. Ce genre de méthode est en général associé à l'extraction des minuties pour augmenter les performances du système, ou utiliser pour classifier les empreintes.

L'archivage dans la base de données permet de caractériser de manière unique la personne.

I.2.4. Le stockage et la phase d'appariement

Pour les systèmes disposant de grosses bases de données, l'identification peut poser problème en temps de calcul si la signature d'entrée doit être comparée avec toutes les signatures présentes dans la base.

C'est pourquoi un processus de *classification* et de *déclassification* est nécessaire pour limiter les temps de recherche.

Lorsqu'une image est stockée, un groupe spécifique lui est attribué en fonction de ses caractéristiques.

Lors de l'identification on désarchive l'ensemble des signatures de la base correspondant au groupe de l'empreinte nécessitant l'identification. Puis chacune des images désarchivées est comparée avec celle de l'utilisateur. Ceci permet de réduire sensiblement les temps de recherche en limitant le nombre d'images à comparer, à condition que les différentes catégories soient judicieusement choisies. Parmi les différentes techniques existantes [21] on

distingue principalement l'approche syntaxique (*l'image est décrite au moyen de règles et de symboles et une analyse grammaticale permet de lui associer une classe*), l'extraction des singularités de l'image (*la position des centres et delta permet de déterminer la classe de l'empreinte*) et l'utilisation des réseaux de neurones.

La phase d'appariement est *l'étape critique* du système, elle reçoit en entrée deux signatures issues de deux acquisitions différentes d'empreinte et renvoie en sortie un résultat binaire indiquant si oui ou non les deux signatures proviennent de la même empreinte. Bien entendu deux empreintes provenant de la même personne ne seront jamais identiques en raison de l'élasticité de la peau, de la présence de poussière, de l'orientation du doigt lors de l'acquisition Ceci est caractéristique des systèmes biométriques. La phase d'appariement va donc calculer le degré de similarité (*taux d'appariement*) entre les deux signatures et *décider* si elles peuvent être considérées identiques en fonction d'une valeur seuil [22].

Bien que les deux empreintes puissent être comparées directement par corrélation [23] la méthode qui a suscité le plus d'intérêt utilise les caractéristiques locales des minuties et consiste en l'appariement basé sur l'alignement d'un motif de points [24] car il est simple en théorie, efficace pour faire face à la fausse information détectée dans les phases précédentes, et rapide par rapport aux autres méthodes. Cet algorithme est divisé en deux processus:

- *L'alignement*: on évalue la transformation géométrique (orientation, translation, homothétie) entre les deux ensembles à traiter et on les aligne suivant cette transformation.
- *L'appariement*: on évalue le nombre d'éléments caractéristiques qui sont alignés (moyennant une certaine marge d'erreurs car un alignement parfait est impossible) et le taux d'appariement est calculé en fonction des correspondances rencontrées.

I.3. Les problèmes posés par l'utilisation de la biométrie

Dans les systèmes biométriques, la phase finale d'appariement estime le degré de similitude (*taux d'appariement*) entre deux fichiers signatures et le compare à un seuil fixé à l'avance, ainsi le résultat n'est jamais fiable à 100% mais s'en approche selon le réglage du seuil [22]. L'évaluation des performances de ces systèmes fait apparaître *deux types d'erreur* [25]:

- Le Taux de Faux Rejets (TFR) correspondant au pourcentage de personnes rejetées par erreur.
- Le Taux de Fausses Acceptations (TFA) correspondant au pourcentage de personnes qui ont été acceptées et qui n'auraient pas du l'être.

La relation liant TFA et TFR est illustrée sur la Figure I-9. On peut voir que plus le TFR est faible et plus le TFA est élevé et inversement. Ces deux éléments dépendent du réglage du seuil qui est le résultat d'un compromis selon le choix de l'application. Pour des applications de haute sécurité (accès au coffre fort de la banque centrale par exemple), on cherchera à limiter au maximum la possibilité d'intrusion, ce qui se traduit par un TFA faible. Au contraire, dans le cadre d'applications médicales, on ne pourra pas se permettre de rejeter une personne par erreur ce qui implique un TFR le plus faible possible.

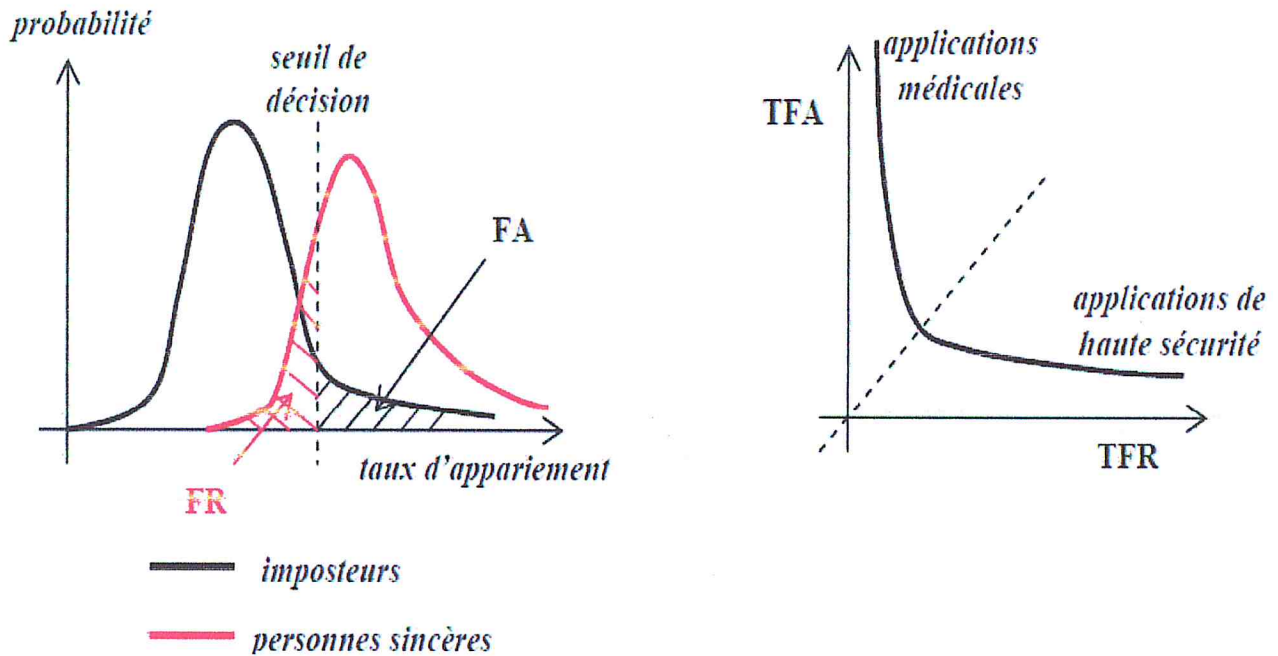


Figure I-8: Liens entre TFR et TFA

Les systèmes biométriques posent aussi des *problèmes de sécurité* car il n'est jamais vérifié que les données biométriques en entrée proviennent de leur réel possesseur. En effet les moyens de flouer un tel système existent [26], et de plus les caractères biométriques ne sont pas secrets. Il est relativement facile de se procurer une photo du visage, une empreinte digitale ou un enregistrement audio d'une personne. Une équipe japonaise a d'ailleurs montré que les systèmes actuels d'acquisition d'empreintes acceptent très bien une fausse empreinte créée à partir d'un moule en gélatine [27]. Le problème vient de l'absence de vérification que la donnée entrée provient bien d'un être vivant.

L'utilisation de la biométrie dans la vie de tous les jours pose également des *problèmes de vie privée* et de libertés [28]. En France l'absence de législation spécifique sur les techniques biométriques peut poser problème et a fait l'objet d'un rapport parlementaire très détaillé [29]. La commission Nationale de l'Informatique et des Libertés [30] a d'ailleurs rendu un avis défavorable sur le stockage en masse de caractéristiques biométriques des personnes face aux risques de réutilisations abusives de ces données.

I.4. Conclusion

Dans ce chapitre nous avons vu l'émergence des méthodes biométriques de reconnaissance et des problèmes qui en découlent. Les caractéristiques des empreintes digitales ainsi que la structure globale d'un système de reconnaissance d'empreintes ont également été décrit; il s'agit à l'heure actuelle de la technique biométrique la plus aboutie.

Dans le chapitre suivant, nous allons étudier la compression ; les principales méthodes ainsi quelques standards seront présentés.

Introduction

La qualité visuelle des images numériques augmente continuellement avec le développement de nouvelles techniques d'affichage (multirésolution, transmission progressive) et de nouvelles technologies d'acquisition (haute définition, nouveau hardware). Cependant, la taille de ces images augmente proportionnellement à leur qualité et leur stockage et transmission constituent donc les enjeux principaux dans le monde numérique. La compression s'impose comme une étape incontournable pour optimiser l'utilisation de ces grands volumes d'informations dans les réseaux informatiques. L'objectif principal de la compression d'images est de réduire la quantité d'information nécessaire à une représentation visuelle fidèle à l'image originale. Nous différencions les schémas de compression selon la perte d'informations. Les méthodes réversibles, section II.2, utilisent uniquement le principe de la réduction de la redondance et n'engendrent pas de perte.

Les méthodes irréversibles, section II.3, définissent une représentation approximative de l'information.

Dans les sections suivantes, nous présentons quelques théories et aspects des nouvelles techniques de compression des images numériques.

II.1 Généralités sur la compression d'images

Une image peut être représentée sous la forme vectorielle (les données sont représentées par des caractéristiques géométriques, comme traits, surfaces, etc.) ou sous la forme d'une matrice de points, bitmap. Dans ces travaux de mémoire, le terme image correspond au type bitmap ou matrice de points.

Une image est une matrice de ($M \times N$) points appelés pixels et à chaque pixel est associé une ou plusieurs valeurs d'intensité qui se combinent pour déterminer la couleur.

II.1.1 Taux de compression et redondance

Le taux de compression soumis à une image est directement proportionnel à la quantité de redondance d'information qu'elle possède [31].

II.1.1.1 Taux de compression

Le taux de compression est utilisé pour mesurer le résultat d'un procédé de compression.

Il est représenté soit comme une formule, équation (a), soit comme un facteur, équation (b). Dans les équations (a) et (b), I_o est la taille de l'image originale en octets et I_c la taille de l'image comprimée. Le taux de compression peut être aussi quantifié par le nombre moyen de bits par pixel (bpp), équation (c). L'élément $Bits_{I_c}$ est le nombre total de bits de l'image comprimée et $Pixels_{I_o}$ est le nombre total de pixels de l'image originale.

$$(a) \quad \sigma = \frac{I_o}{I_c} \qquad (b) \quad \sigma = (I_o/I_c) : (1) \qquad (c) \quad \sigma = \frac{Bits_{I_c}}{Pixels_{I_o}}$$

II.1.1.2 Redondance

Une image numérique présente la particularité de posséder des corrélations importantes entre les pixels voisins. Cette corrélation est vue comme une redondance des informations pertinentes. La redondance peut être de deux natures : *la redondance spatiale* qui apparaît directement entre les pixels voisins de l'image originale et la *redondance spectrale* qui est liée aux fréquences et qui est acquise avec les transformations de domaines. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle.

II.1.2 Critères psychovisuels et compression

Les méthodes de compression sans perte ne causent aucun problème visuel car elles sont totalement réversibles. Par contre, les procédures de compression avec pertes diminuent la qualité de l'image. Plus le taux de compression est important, plus les distorsions apparaissent dans l'image. Le point critique est la définition de la quantité de distorsions par rapport à la qualité de l'image. Le SVH (Système Visuel Humain) possède des caractéristiques particulières qui doivent être prise en compte. Notre œil est capable de distinguer environ sept millions de couleurs [34]. Quand nous regardons une image notre système visuel doit résoudre beaucoup de contraintes : perception 3D, ombres, objets cachés, etc. En fait, le SVH tente de donner un sens visuel à chaque objet. Notre perception est influencée par ce que nous nous attendons à voir, c'est le cas des illusions optiques par exemple.

La perception de la couleur est influencée fortement par la saturation et la luminance. La saturation est la quantité de blanc ajoutée à une couleur, et la luminance est la mesure de lumière réfléchiée par un objet. L'œil humain possède des sensibilités différentes suivant l'orientation des contrastes (Il a la propriété de distinguer deux régions distinctes dans une image, à l'aide de la différence de couleur et de luminosité.). Il est beaucoup plus sensible à la luminance qu'à la chrominance. Les systèmes de compression avec pertes changent généralement d'espace couleur. Il est conseillé de ne pas dégrader beaucoup la composante de luminance. La norme JPEG, décrite section 2.5, par exemple, sous-échantillonne les deux composantes de chrominance, et ne change pas la composante de luminance [42].

II.1.2.1 Contours et Texture

La perception des contours fait partie des fonctions essentielles du SVH. Nous délimitons mentalement les objets qui sont dans une image grâce à leurs contours, et nous sommes très sensibles à la dégradation de ces contours. La théorie des formes explique leur importance pour la perception correcte des objets partiellement cachés. La fermeture des contours pour produire une forme visuelle connue par notre mémoire est essentielle [34].

II.2 Codage sans perte

La compression sans perte ou codage entropique ou codage réversible permet de retrouver la valeur exacte du signal comprimé lorsqu'il n'y aucune perte de données sur l'information d'origine. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans perte crée des "mots-codes" à partir d'un dictionnaire statique ou d'un dictionnaire construit dynamiquement [32]. Ces processus s'appuient sur des

informations statistiques de l'image. Les codes statistiques les plus répandus sont le codage d'Huffman et le codage arithmétique. Le codage statistique permet de s'approcher au mieux de l'entropie [31]. Ils ont pour principe d'associer aux valeurs les plus probables les mots binaires les plus courts.

II.2.1 Codage d'Huffman

Huffman [32] a suggéré une méthode statistique qui permet d'attribuer un mot-code binaire aux différents symboles (pixel) à compresser. La probabilité d'occurrence du symbole dans l'image est prise en compte en attribuant aux plus fréquents des codes courts, et aux plus rares des codes longs, *VLC - Variable Length Coding*. La suite finale de pixels codés à longueurs variables sera plus petite que la taille originale. Le codeur Huffman crée un arbre ordonné à partir de tous les symboles et de leur fréquence d'apparition. Les branches sont construites récursivement en partant des symboles les plus fréquents. Le code de chaque symbole correspond à la suite des codes le long du chemin allant de ce caractère à la racine. Plus le symbole est profond dans l'arbre plus la quantité de bits pour le représenter est importante. La figure 2.1 présente un exemple de codeur d'Huffman. Le tableau de gauche montre que nous avons au total 21 symboles qui sont représentés en octets équivalant à 168 bits. Après la construction de l'arbre d'Huffman nous pouvons constater un taux de compression de :

$$\sigma = \frac{45}{168} \text{ de } 26,79\%.$$

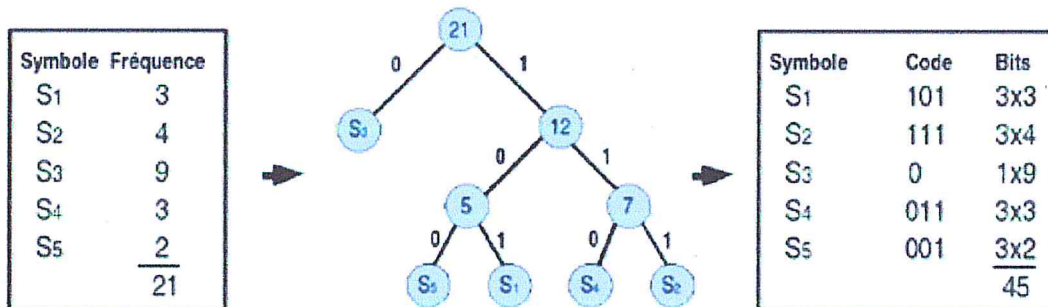


Figure 2-1: Algorithme d'Huffman

II.2.2 Codage arithmétique

Le codage arithmétique (CA) [33] est un codage statistique qui attribue à une suite de symboles une valeur réelle. Il consiste à découper l'intervalle des réels $[0, 1[$ en sous-intervalles, dont les longueurs sont fonctions des probabilités des symboles. Le codage arithmétique n'attribue pas un code à chaque symbole comme Huffman et les autres codages par blocs, mais un code au message tout entier. Les tableaux suivants présentent un exemple de codage arithmétique avec le message AAOEU.

Alphabet	Probabilité	Probabilité Cumulée	Partition initiale	Mes-sage	Gauche G	Taille T	Droite D
A	0,2	0,2	[0 0,2)	A	0,0000	0,2000	0,2000
E	0,4	0,6	[0,2 0,6)	A	0,0000	0,0400	0,0400
I	0,1	0,7	[0,6 0,7)	O	0,0280	0,0080	0,0360
O	0,2	0,9	[0,7 0,9)	E	0,0296	0,0032	0,0328
U	0,1	1,0	[0,9 1,0)	U	0,0325	0,0003	0,0328

TAB 2-1 : Codage Arithmétique

Soit l'alphabet {A, E, I, O, U} avec les probabilités {0,2 0,4 0,1 0,2 0,1}. Le codage arithmétique est fait à partir de l'intervalle initial [0, 1[et au fur et à mesure du codage, la longueur de l'intervalle diminue en tenant compte du sous-intervalle précédent.

Nous nous servons des formules :

$$(G = G_I^P + G_M * T_I^P) \quad \text{et} \quad (T = T_I^P * T_M)$$

pour construire le nouvel intervalle où les lettres signifient : G-gauche, T-taille, M-message et P-précédant.

Le premier symbole **A** du message réduit l'intervalle initial à [0 0,2[. Le deuxième symbole **A** du message réduit ce dernier intervalle à [0 0,04[(1/5 de l'intervalle précédent).

Le symbole **O** réduit l'intervalle à [0,028 0,036[. Le symbole **E** diminue l'intervalle à [0,0296 0,0328[. Enfin, le symbole final **U** réduit à [0,03248 0,0328[. Finalement, tout réel dans l'intervalle [0,03248 0,0328[codera le message AAOEU. Le codage arithmétique est présent dans la norme JPEG (dans les modes Extended DCT-based processes et Lossless processes) et JPEG2000.

Les méthodes de codage statistiques construisent les mots-codes à partir d'un dictionnaire prédéfini, basé sur les statistiques de l'image elle-même. Ce dictionnaire est indispensable pour le décodage. De nouvelles études et améliorations pour cette approche ont été proposées, nous citons les travaux [34, 35, 36].

Les deux méthodes que nous allons présenter, *codage par substitution*, n'exigent pas de connaissance a priori de l'image, comme les probabilités d'apparition des pixels par exemple. Elles construisent des dictionnaires dynamiques dont les mots-codes créés sont indépendants de la source.

II.2.3 L'algorithme LZW

Lempel et Ziv [37] ont présenté un schéma (LZ77) qui est à la base de tous les algorithmes à dictionnaire dynamique utilisés actuellement. Welch a amélioré leur algorithme et a déposé un brevet en créant l'algorithme LZW qui génère un dictionnaire dynamique qui contient des motifs du fichier [38]. L'utilisation d'un dictionnaire dynamique a réglé le problème de le transmettre a priori pour les procédés de compression et décompression. Il est basé sur la multiplicité des occurrences de séquences de symboles.

Son principe consiste à substituer des motifs par un code d'affectation en construisant au fur et à mesure un dictionnaire. Celui-ci est initialisé avec les valeurs de la table ASCII. Chaque octet du fichier est comparé au dictionnaire. S'il n'existe pas, il est ajouté au dictionnaire. L'algorithme LZW fait partie du format d'image, aussi breveté, GIF - (Graphics Interchange Format).

II.2.4 Codage par plage

Le codage par plage ou RLE Run Length Encoding est recommandé lorsque nous observons des répétitions de symboles consécutifs. Il est utilisé par de nombreux formats d'images (BMP, TIFF, JPEG) [31]. L'idée est de regrouper les pixels voisins ayant la même couleur. Chaque groupement définit un couple de valeurs $P = (\text{plage}, n)$ où plage est le nombre de points voisins ayant la même valeur, et n est cette valeur. Le RLE est d'autant plus performant que les groupements sont étendus, il n'est pas applicable dans tous les cas. Il est recommandé pour les images avec de larges zones uniformes. La compression d'une image peut être effectuée de manière adaptative : dans les régions uniformes le RLE est appliqué, et dans les zones non uniformes des règles particulières sont créées. Par exemple, au moins trois éléments se répètent consécutivement alors la méthode RLE est utilisée, sinon un caractère de contrôle est inséré, suivi du nombre d'éléments de la chaîne non compressée. D'autres caractères de contrôle peuvent aussi être utilisés pour définir la fin de ligne ou la fin de colonne.

II.2.5 Codage par prédiction linéaire

Les algorithmes qui utilisent le codage par prédiction exploitent la redondance spatiale.

Il s'agit de prédire la valeur d'un pixel en fonction de la valeur des pixels voisins et de ne coder que l'erreur de prédiction. Le gain en compression est accompli par la variation faible entre pixels voisins, sauf pour les pixels situés sur les contours. Le voisinage peut être défini selon sa connexité (4-connexité ou 8-connexité) ou selon l'ordre du parcours choisi pour accéder aux pixels voisins. L'une des techniques de prédiction la plus simple est la DPCM (Differential Pulse Code Modulation) [38]. Cette technique effectue une prédiction à base d'une combinaison linéaire des valeurs des pixels voisins. Une version adaptative, ADPCM, qui utilise différentes formes de prédiction et de voisinage selon le contexte et le contenu de l'image a été présentée par Kyung et al. [39]. Récemment, Babel et al. [40] ont proposé un codage progressif et multirésolution LAR - (Locally Adaptive Resolution). Il s'agit d'un codeur qui associe le DPCM à une décomposition multi-couches suivi d'une transformée Mojette (C'est une transformation qui projette des informations disposées en 2D sur des vecteurs 1D.). La profondeur de la décomposition détermine le type de compression, avec pertes ou sans perte pour le huitième niveau.

II.3 Codage avec pertes

Les méthodes avec pertes (lossy) ou irréversibles sont des méthodes qui tirent parti d'une corrélation (ou redondance) existante dans l'image. L'information perdue est due à l'élimination de cette redondance, ceci rend possible une compression plus importante. La perte d'information est toujours discutable et nous nous posons alors la question de la limite acceptable. Cette limite est définie par le type d'application, comme les images médicales ou satellites par exemple. La quantification est un des mécanismes utilisés dans les algorithmes de compression, qui produit des pertes d'information.

II.3.1 Quantification

La quantification fait partie de plusieurs méthodes de compression d'image. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image.

II.3.1.1 Scalaire

La quantification scalaire SQ - (Scalar Quantization) est une procédure qui associe à une variable continue X une variable discrète x . Pour cela on associe à x la valeur quantifiée : $x_q = Q(X)$ où Q est une fonction (non linéaire) de quantification de $\mathbb{R} \rightarrow \mathbb{Z}$ [41].

La quantification scalaire utilisée, en pratique, dans les images sont des quantifications basées en zone morte dans lesquelles l'intervalle de quantification, est centré à l'origine et est de taille multiple de la taille des autres intervalles de quantification [42].

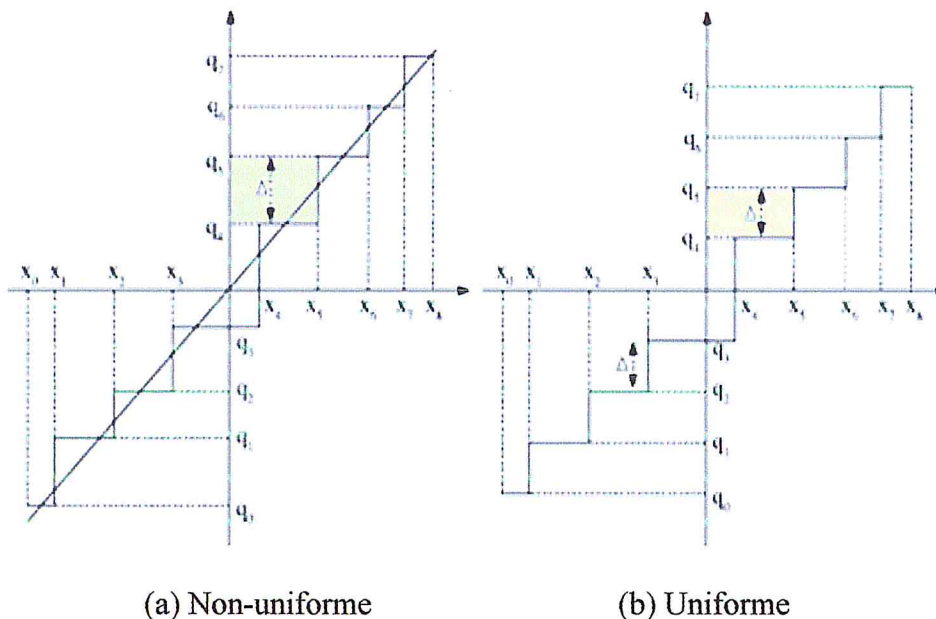


Figure. 2-2 : Quantification scalaire

II.3.1.2 Vectorielle

La quantification vectorielle VQ - (Vector Quantization) a été développée par Gersho et Gray [43] et elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique [44]. Le principe de la quantification vectorielle est issu du travail de Shanon qui montre qu'il est toujours possible d'améliorer la compression de données en codant non pas des scalaires, mais des vecteurs. Un quantificateur vectoriel Q associe à chaque vecteur d'entrée $X_i = (x_j, j = 1 \dots k)$ un vecteur $Y_i = (y_j, j = 1 \dots k) = Q(X_i)$, ce vecteur Y_i étant choisi parmi un dictionnaire (code-book) de taille finie. La VQ produit de meilleurs résultats que la SQ, néanmoins la VQ nécessite un codage complexe.

II.3.2 Codage prédictif avec pertes

Il existe des techniques qui exploitent la redondance spatiale, cependant la prédiction est faite par approximation. Ces algorithmes ont comme objectif de rechercher un modèle de représentation le plus adéquat de l'information à coder afin d'obtenir un coût de codage minimal. L'idée est de coder l'erreur de prédiction au-dessus d'un seuil. Ce seuil peut être défini par rapport à la qualité de l'image ou le niveau de compression espéré. Quelques résultats sont présentés dans les travaux [45,46].

II.3.3 Codage par transformation

Les méthodes qui reposent sur cette technique utilisent des transformations pour produire une décorrélation des redondances spectrales [38]. Les pixels passent d'un espace où ils sont fortement corrélés dans un autre espace où leur corrélation est moindre. Lors de chaque transformation, le signal d'origine est remplacé par sa représentation dans un autre domaine. Dans divers algorithmes, cette transformation d'espace est accompagnée d'une quantification et d'un codage entropique pour accomplir la compression de l'image. Ceci est le cas des standards de compression : l'algorithme JPEG (section II.5.1.1) qui utilise la transformation type DCT et l'algorithme JPEG2000 (section II.5.1.3) qui utilise la transformation en ondelettes DWT [42].

II.4 Domaine spatial

Les méthodes de compression d'images dans le domaine spatial exploitent la redondance entre un pixel et son voisinage, ou entre certaines régions de l'image.

II.4.1 Quadtree

La technique de décomposition par quadtree est basée sur une approche récursive d'un codage arborescent. Une image hétérogène de taille $2n \times 2n$ est alors divisée en quatre sous-régions, de taille $2n-1 \times 2n-1$. Ce processus est refait jusqu'à ce que chacune des régions soit déclarée homogène selon un critère choisi. Certains travaux proposent d'optimiser ce compromis en découpant l'image en blocs de tailles variables [47,48]. Des zones considérées homogènes seront découpées en grands blocs alors que des zones texturées ou contenant des contours seront découpées en blocs plus petits comme le montre la figure 2.3.

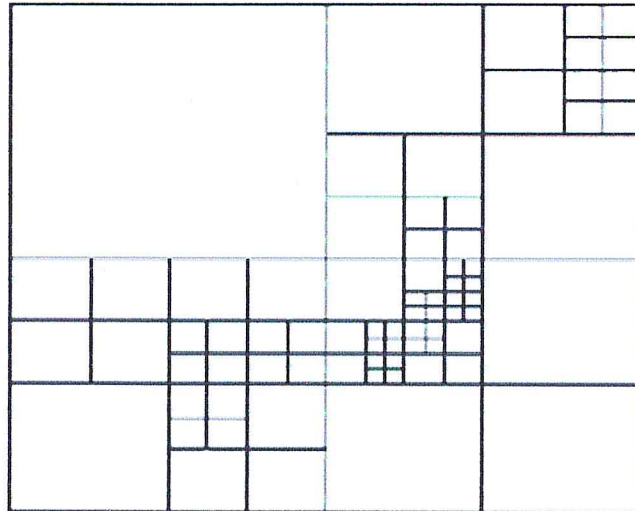


Figure 2-3 : Décomposition en quadtree

II.4.2 Décomposition en plans binaire

Cette technique est principalement employée pour les images en niveau de gris [37]. L'idée est de décomposer l'image en huit images binaires, une pour chaque bit de niveau de gris, en commençant par les bits de poids les plus forts (figure 2.4).

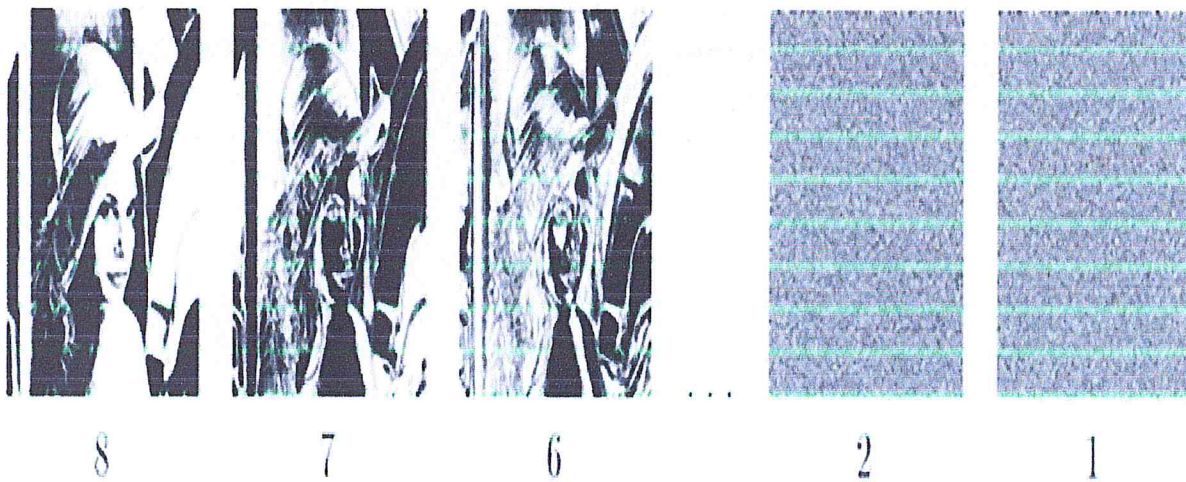


Figure 2-4 : Décomposition en plans binaires des bits de poids forts jusqu'aux bits de poids faibles

Sur chaque image binaire sont lancées des procédures particulières pour effectuer la compression. L'image binaire représentant le bit de poids le plus fort, première image (8), apporte le plus haut taux de compression, tandis que la compression sur l'image binaire de poids le plus faible, dernière image (1) a quasiment une compression nulle.

Maniccamam et Bourbakis [49] ont présenté une méthode de compression sans perte basée sur la décomposition de l'image en plans binaires et sur la méthodologie SCAN (est un langage formel basé sur l'accès spatial en deux dimensions). Dans leur approche, chaque plan

binaire est également décomposé en régions, et chaque région est parcourue selon un ensemble de chemins différents. Cet ensemble de chemins est composé de 32 chemins standards de plus d'autres chemins sont générés selon le langage SCAN. Pour chaque chemin de cet ensemble est réalisé un codage RLE pour déterminer le taux de compression optimal. La compression de l'image est déterminée pour l'ensemble des compressions de chaque région.

II.4.3 Fractales

La compression par fractale est une technique de compression avec pertes encore peu utilisée. Un fractal est une structure géométrique qui se reproduit, dans une boucle infinie, par transformation affine (translation, rotation et mise à l'échelle). Cette structure se refait à toutes les échelles de forme réduite et légèrement déformée. La compression par fractale est basée sur le principe qu'il existe des similarités entre différentes régions isolées d'image. Elle exploite les récurrences des motifs qui, après quelques traitements, peuvent permettre une compression. La figure 2.5 présente un exemple d'exploitation des motifs. Dans la compression par fractales nous distinguons les travaux [50,51].



Figure 2-5 : Compression par Fractales

II.5 Domaine fréquentiel

Les techniques de compression dans le domaine fréquentiel s'appuient sur une transformation de l'image vers un nouvel espace de représentation d'énergie fortement décorrélée. Cette décorrélation provoque une nouvelle représentation de l'image par la redistribution de l'énergie dans un nombre restreint de coefficients transformés. Cette énergie de l'image transformée est distribuée sous la forme de tranches énergétiques de basse, moyenne et haute intensités. Les transformations d'espace les plus courantes sont la DCT et la DWT.

II.5.1 Les standards

Il existe plus d'une cinquantaine de types de formats d'image [31]. Pour chacun d'entre eux la structuration des données et les attributs sont différents. La standardisation d'un format d'image permet de régler l'utilisation, la divulgation et la production de logiciels et de hardware compatibles avec le format standard. Le format standard JPEG est le format d'image le plus populaire, et il est devant la scène depuis quelques années. Son successeur, le JPEG2000, semble s'établir dans le domaine de l'image numérique. Le JPEG2000 possède des fonctionnalités supplémentaires par rapport au format JPEG. Cependant, la plupart des appareils numériques (appareils photos, caméscopes, téléphones portable, etc) et les logiciels qui capturent et traitent les images sont au format JPEG.

II.5.1.1 JPEG

Le comité Joint Photographic Expert Group a été créé en 1986 par la jonction (Joint) de plusieurs groupes qui travaillaient sur la photographie. Ce comité a produit la norme de compression d'images photographiques qui a été standardisée (ISO/IEC/10918-1/ 1994) et a reçu son nom JPEG. Il est devenu le format le plus populaire très rapidement parce qu'il a été conçu avec différentes contraintes [42] :

- L'algorithme JPEG doit être implémentable sur une grande variété de types de CPU (unité centrale de calcul) et sur des cartes plus spécialisées (appareil photo numérique et téléphone portable par exemple).
- Il doit pouvoir compresser efficacement tout type d'images réelles (images photographiques, médicales) avec pertes et sans perte.
- Il possède quatre modes de fonctionnement : séquentiel (*baseline*), progressif (*extended DCT-based*), sans perte (*Lossless*), hiérarchique *hierarchical*.

Entre les 4 modes de compression de la norme JPEG, le séquentiel ou *baseline* est le mode principal le plus répandu. Il est basé sur la transformation DCT, quantification scalaire et le codage d'Huffman sur pixels de 8 bits par plan de couleur.

La figure 2.6 expose une synthèse du mode séquentiel.

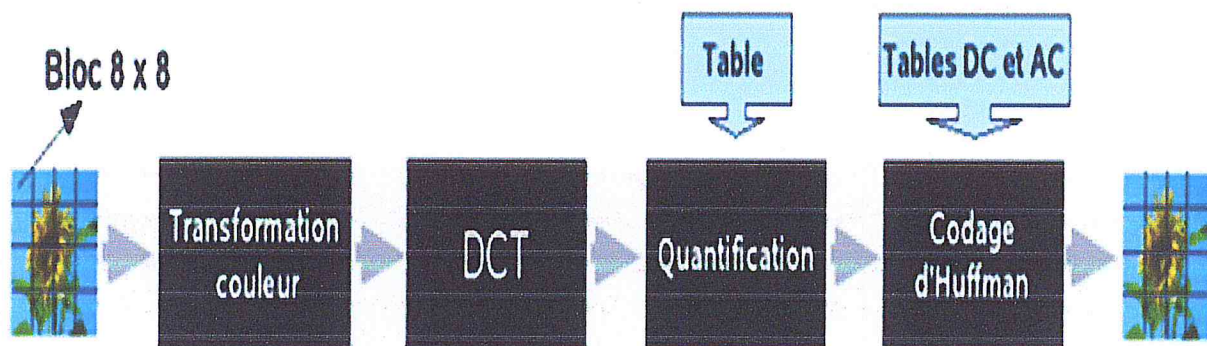


Figure 2-6 : Compression JPEG

➤ **Transformation d'espace couleur**

Tout d'abord, l'image originale est soumise à un changement d'espace couleur YCbCr (Le Y est la luminance, Cb et Cr sont les chrominances). Les informations de chrominance, les plans Cb et Cr, sont sous-échantillonnées.

Après cette transformation, les deux plans auront deux fois moins de lignes et de colonnes.

Cette opération de sous-échantillonnage est fondée sur le principe que le SVH ne peut discerner des différences de chrominance au sein d'un carré de 2×2 points. Après le sous-échantillonnage, chaque plan Y, Cb et Cr est traité de la même manière selon le schéma fonctionnel présenté figure 2.6. Tout d'abord, ils sont découpés en blocs de 8×8 pixels.

➤ **DCT**

Chaque bloc 8×8 est soumis à une transformation par DCT. Le premier coefficient de la DCT, le DC, est proportionnel à la moyenne des valeurs du bloc. Les 63 autres coefficients sont appelés AC. Ce nouveau domaine transformé permet une décorrélation très forte de l'information. Sur ce bloc de coefficients, les énergies sont groupées en basse (zone homogène de l'image originale), moyenne et haute fréquences (zone texturée ou zone de contour). Avec la DCT, chaque colonne est une fonction cosinus de fréquence différente. La variance est alors concentrée sur les composantes de basse fréquence, et les composantes de haute fréquence seront annulées par quantification.

➤ **Quantification**

La compression avec pertes est faite dans l'étape de **quantification** qui est réalisée à l'aide d'une matrice Q de quantification de 8×8 éléments (figure 2.6).

L'image subit des distorsions selon le niveau de compression désiré. Chaque coefficient DCT est divisé par la valeur correspondante dans Q et le résultat est arrondi à l'entier le plus proche. L'acuité du SVH est plus faible à des hautes fréquences et plus sensible aux basses fréquences. Quelques tables standards, pour la quantification, ont été générées grâce à une série de caractéristiques psychovisuelles.

Les valeurs prennent donc en compte cette caractéristique et introduit majoritairement de la distorsion dans les hautes fréquences. Cependant, il est tout à fait possible de personnaliser la table [52].

➤ **Codage d'Huffman**

Le codage entropique dans le mode séquentiel est un codage du type RLE. Après la quantification un grand nombre de coefficients sont nuls ou très proches de zéro.

Le coefficient DC (composante continue) est codé séparément par rapport aux AC.

Les coefficients AC sont parcourus en zigzag, figure 2.7, et sont codés par des couples (HEAD),(AMPLITUDE). L'entête HEAD contient des contrôleurs qui seront utilisés pour accéder aux tables d'Huffman. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude du coefficient AC non nul. La structure HEAD varie en fonction du type de coefficient. Pour les AC elle est composée de (RUNLENGTH, SIZE), alors que pour les DC elle est composée seulement de la taille SIZE. Les coefficients DC transportent une information visible importante et une corrélation

locale significative. Ils sont hautement prédictibles, ainsi JPEG traite les coefficients DC séparément des 63 coefficients AC. La valeur des composants DC est importante et variée, mais est souvent très proche de celle de ses voisins. La seule valeur qui est donc encodée est la différence DIFF entre le coefficient DC_i quantifié du bloc courant et le précédent DC_{i-1}.

La figure 2.7 exhibe un bloc 8×8 de coefficients DCT quantifiés et le chemin en zigzag utilisé par le codage d'Huffman.

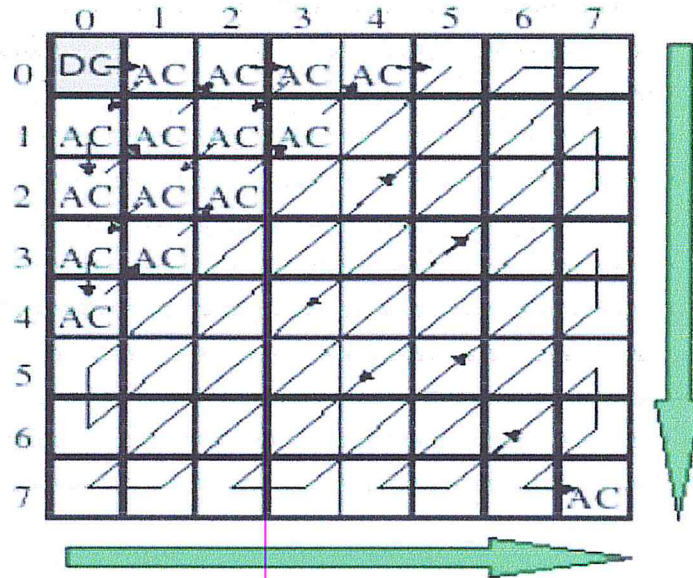


Figure 2-7 : Bloc DCT et parcours zigzag

II.5.1.2 JPEG-LS - [ISO/IEC/14495-1/ 1999]

La compression JPEG en mode sans perte n'est pas optimisée. Les objectifs du comité étaient de concevoir un mode réversible permettant une compression de l'image à 50%. Malheureusement, un code réversible est quasiment impossible avec l'utilisation de la DCT. Les erreurs d'arrondis dues à la précision limitée de calcul sont toujours présentes.

Le nouveau standard de compression sans perte, le JPEG-LS est basé sur une variation de la méthode LOCO-I - (*low complexity lossless compression method*) [53]. Dans JPEG-LS le procédé de compression est composé de trois parties : la *prédiction* de la valeur du pixel qui est faite par rapport aux pixels voisins en utilisant l'approche de prédiction MED (*Median Edge Detection*) ; la *détermination d'un contexte*. Ce contexte représente l'environnement du pixel à coder et ses voisins. L'idée est de prendre le meilleur environnement qui affine la prédiction avant le codage et de réduire le nombre de paramètres de l'erreur de prédiction ; le codage de l'erreur de prédiction dans l'approche est de réinsérer l'erreur de prédiction dans le système puis de la comparer à d'autres mesures d'erreur.

Le JPEG-LS possède une option de haut taux de compression, mais cette option est quasiment sans perte (*near lossless*). Les observations expérimentales montrent que pour des taux supérieurs à 1,5 bpp, le JPEG-LS en mode *near lossless* donne de meilleures performances que JPEG [53].

II.5.1.3 JPEG2000 - [ISO/IEC/15444-1/ 2000]

Le JPEG2000 [42] remplace le JPEG comme le format standard pour la compression des images. Il a été réalisé dans la perspective de répondre aux exigences des nouvelles applications les plus diversifiées, comme la multirésolution par exemple.

La compression JPEG2000 est composée de plusieurs étapes selon les schémas avec pertes et sans perte. Tout d'abord, *un changement d'échelle* est effectué dans chaque composante couleur RGB, l'échelle est changée de (0, 255) à l'échelle (-128, 127) par une simple soustraction de 128 de chaque valeur. Après le changement d'échelle, l'image est soumise à *une transformation de plans couleurs*, facultative, de RGB à YCbCr. Cette transformation peut être réversible (sans perte) ou irréversible (avec pertes).

Chaque plan chromatique de l'image est découpé en petites images appelées *tuiles*. Chaque tuile est considérée comme une image et est traitée de façon indépendante.

Du fait de la complexité du mécanisme du JPEG2000, la décomposition de l'image en tuiles rend possible l'application du JPEG2000 sur des images de taille importante.

La transformation de domaine est faite à l'aide d'une décomposition en ondelettes par schéma *lifting* [54]. Chaque tuile de chaque composante subit des transformations selon les types compressions réversible et irréversible. Pour la compression réversible la norme utilise la paire (5,3) de LeGall. Les coefficients des filtres d'analyse sont entiers et composés de 5 coefficients pour le filtre passe-bas et de 3 pour le passe-haut.

Pour la compression irréversible est utilisée la paire (9,7) de Daubechies qui utilise des coefficients réels, avec 9 coefficients pour le passe-bas et 7 pour le passe-haut. Les lignes et les colonnes sont récursivement décomposées générant quatre sous-bandes pour chaque niveau de décomposition. Ceci produit des sous-bandes LL (basses fréquences horizontales et verticales), LH (basses fréquences horizontales et hautes fréquences verticales), HL (hautes fréquences horizontales et basses fréquences verticales) et HH (hautes fréquences horizontales et verticales).

Ensuite, une étape de quantification est faite selon le taux de compression. La quantification est faite pour rendre nuls les coefficients les plus faibles et faciliter le codage des autres. Chacune des sous-bandes peut être quantifiée avec des étapes différentes.

Plusieurs techniques permettant la quantification des sous-bandes ont été proposées, linéaire, non-linéaire, avec masquage en fonction des voisins ou codage en treillis (TCQ), mais seule la quantification linéaire a été retenue par la norme.

Avant le codage entropique, les coefficients quantifiés sont divisés en plusieurs blocs appelés code-blocs de taille 64x64 ou 32x32. Le codage entropique du JPEG2000 est du type arithmétique adaptatif avec contexte, EBCOT - (*Embedded Block Coding with Optimized Truncation*) [42]. Le codage d'un bloc consiste à parcourir les coefficients du code-block par plan de bits, du bit de poids le plus fort (MSB) au bit de poids le plus faible (LSB). Les bits sont séparés en trois groupes en fonction de leur voisinage.

Ensuite, ils sont codés en trois passes (*coding passes*). Le codage débute lorsqu'un plan de bits devient significatif (*signifiant*).

La résistance aux erreurs est une caractéristique particulière du JPEG2000. Après le codage entropique, plusieurs caractères de contrôle (*segment marks, resynchronising marks*) sont insérés dans le flux de bits. Cette démarche est faite pour synchroniser les informations, limiter la taille du segment et éviter la propagation des erreurs.

Une autre fonctionnalité importante du JPEG2000 est la compression par région d'intérêt (ROI). Ceci permet d'avoir des taux de compression différents dans certaines régions de l'image. Les zones importantes peuvent être compressées quasi sans pertes et les zones moins importantes avec un fort taux de compression.

Malgré ces nombreuses fonctionnalités, le JPEG2000 possède quelques inconvénients. Il nécessite entre deux et six fois plus de cycles de CPU que JPEG et il n'est pas indiqué pour les machines avec faibles ressources comme les appareils photos numériques par exemple. L'algorithme JPEG est beaucoup moins complexe et il peut être implémenté en hardware.

Conclusion

Dans ce chapitre, nous avons présenté plusieurs techniques de compression d'images sans perte et avec pertes. Tout d'abord, nous avons donné quelques concepts et mesures pour la compression d'images. Nous avons vu qu'un système de compression d'images se décompose fondamentalement en : acquisition, prétraitement, quantification et codage entropique. D'une manière générale, toutes les approches énumérées précédemment utilisent les corrélations entre pixels voisins dans l'image. Ces corrélations sont le fondement de la compression car elles sont liées à la notion de redondance.

Nous avons vu qu'après un changement de domaine, les énergies sont groupées et que la transformation est généralement accompagnée d'une quantification. La quantification rend les composantes de hautes fréquences nulles et celles-ci présentent alors des plages de valeurs nulles qui suggèrent l'utilisation de codage à longueur variable.

Enfin, la connaissance des caractéristiques du SVH est très importante pour conserver un rapport qualité/compression correct et pour augmenter la compression par une pondération psychovisuelle.

Le chapitre suivant traite de la compression selon le format 'Wavelet Scalar Quantization' utilisé dans la conception de notre système.

Introduction

Cela fait maintenant quelques années que les ondelettes se sont imposées comme une technique d'intérêt pour de nombreux problèmes de traitement du signal. En particulier, dès les premiers travaux de S. Mallat, on a pressenti que les ondelettes pouvaient s'appliquer avec succès au codage d'images et notamment pour les normes de compression comme la norme Wavelet Scalar Quantization.

Cette impression fut confirmée très tôt grâce à la reconnaissance du lien très fort existant entre schéma d'implantation par ondelettes et bancs de filtres, que ce soit en une ou deux dimensions. Ainsi, du point de vue de sa structure algorithmique, la Transformation en Ondelettes Discrètes (TOD) est identique à un banc de filtres itéré sur la branche passe-bas à chaque étape de décomposition, ce qui lui donne son caractère «multi-résolution». La théorie des bancs de filtres, antérieure à celle des ondelettes, avait déjà été appliquée avec succès dans des schémas de décompression par transformée, et la transformée en ondelettes discrètes s'inscrit donc parfaitement dans ce cadre.

III.1 Les Ondelettes

III.1.1 Fréquence locale

Les ondelettes ont été introduites il y a presque 50 ans pour combler une lacune entre deux modes extrêmes de représentation du signal : la représentation d'une fonction par son graphe classique de \mathbb{R} dans \mathbb{R} , ce qui correspond à une décomposition sur la base continue des distributions de Dirac [60]:

$$f(t) = \int_{\mathbb{R}} f(u)\delta(t - u)du$$

et la représentation dans la base de Fourier :

$$f(t) = \int_{\mathbb{R}} \hat{f}(\omega)e^{i\omega t}d\omega$$

La première représentation donne une information précise en temps : la valeur $f(t)$ indique l'intensité du signal au temps t . En revanche l'information fréquentielle est nulle. La valeur en un point de f ne donne aucune information sur le « contenu fréquentiel » du signal f . A l'inverse, la représentation de Fourier donne une information très précise en fréquence, mais ne donne aucune information temporelle. Par analogie, supposons que la courbe f représente un son. La représentation temporelle f permet de lire la localisation temporelle du signal, c'est-à-dire : ici on entend un son, là on n'entend rien, sans que l'on puisse savoir quelle est la hauteur du son, ou bien les notes qui composent un accord. En revanche, la représentation fréquentielle permet de dire : cet air de musique contient un *la*, un *si*, un *ré* et pas de *sol*, mais cette représentation ne permet de dire quand ces notes (de durée finie) sont jouées.

Naturellement, chacune de ces représentations contient les informations de l'autre, puisque la transformation de Fourier permet de passer de l'une à l'autre. Néanmoins, à chaque fois, seul un type d'information est explicitée.

On peut reformuler les observations ci-dessus en disant que les distributions $\delta(\cdot - t)$ sont très localisées en espace et très peu en fréquence, et qu'à l'inverse les $t \mapsto e^{i\omega t}$ ont des résolutions spatiale et fréquentielle respectivement nulle et infinie. La question qui vient naturellement est la suivante : existe-t-il une représentation dans laquelle on puisse lire une information mixte, comme « à tel instant, on entend un la et un do » ? Morlet et Gabor ont cherché à concevoir des fonctions de base qui se situent à mi-chemin entre ces extrêmes, c'est-à-dire qui ont à la fois une bonne localisation fréquentielle et une bonne localisation spatiale.

Une limite théorique dans cette perspective est bien connue : c'est l'inégalité de Heisenberg. Soit une fonction de base f , qu'on suppose de norme L2 égale à 1[61]:

$$\int |f(t)|^2 dt = 1.$$

On définit le centre $c(f)$ et la largeur $\Delta(f)$ d'une telle fonction par

$$c(f) = \int t|f(t)|^2 dt$$

$$\Delta(f) = \sqrt{\int (t - c(f))^2 |f(t)|^2 dt}$$

L'inégalité de Heisenberg est une inégalité fondamentale qui s'écrit pour toute fonction f de norme 1 :

$$\Delta(f)\Delta(\hat{f}) \geq \frac{1}{2}$$

Si on appelle largeur fréquentielle d'une fonction f la largeur de sa transformée de Fourier \hat{f} , cette inégalité interdit donc d'avoir une fonction avec des largeurs temporelle et fréquentielle toutes deux aussi petites que l'on veut.

On connaît également les fonctions qui réalisent le minimum de cette limite théorique: ce sont les fonctions gaussiennes translatées et modulées :

$$Ae^{-(t-t_0)^2/2\Delta t^2} e^{i\omega_0 t}$$

où A est un coefficient de normalisation tel que la fonction ait une norme L2 égale à 1, qui dépend de Δt . Pour de telles fonctions, et seulement pour ces fonctions-là, l'inégalité (H) devient une égalité. Ces fonctions ont été appelées ondelettes de Gabor.

III.1.2 Représentations temps – fréquence et temps – échelle

A une telle fonction, on associe un pavé temps-fréquence, c'est-à-dire un rectangle dans le plan (t, ω) centré en $(c(f), c(f'))$ et de dimensions $\Delta(f) \times \Delta(f')$. Ce pavé est une représentation intuitive de la couverture en temps et en fréquence d'une fonction. On associe également à une base un pavage du plan temps-fréquence, qui est un recouvrement du plan (t, ω) par des rectangles de couverture des fonctions de base. Si le centre des boîtes de Heisenberg est fixé comme le point des centres spatiaux et fréquentiel de la fonction, leurs dimensions sont en général choisies de telle manière que les boîtes forment une partition du plan temps-fréquence. Cette représentation a un aspect arbitraire, d'autant qu'aucun résultat ne lie le fait qu'une famille soit une base au fait que les pavés temps-fréquence de la famille recouvrent le plan [62].

Le pavage temps-fréquence correspondant par exemple aux bases de fonctions de Dirac et de fonctions de Fourier sont des pavages par des rectangles infiniment fins et allongés qui sont représentées schématiquement dans la figure 3.1.

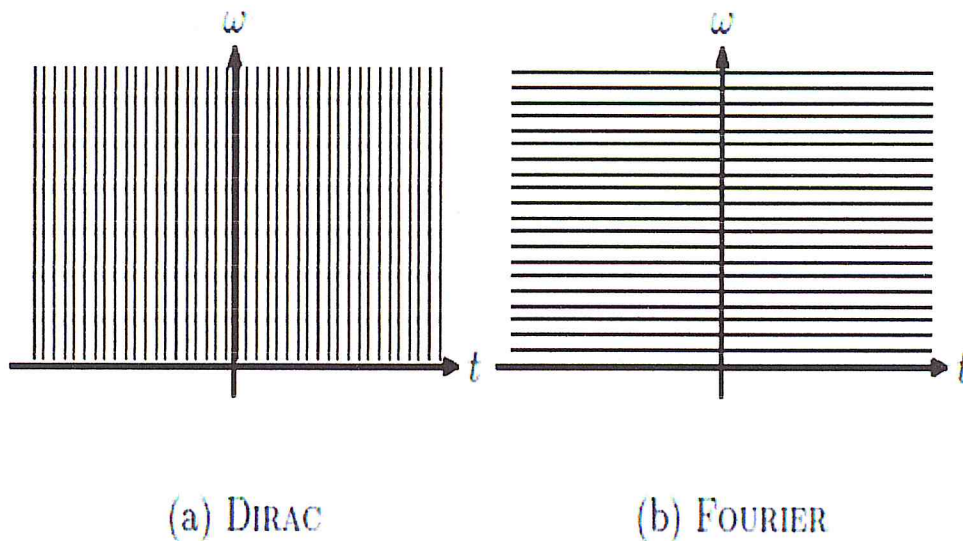


Figure. 3-1 : Pavages temps-fréquences associés aux représentations de Dirac et de Fourier

Pour représenter un signal comme combinaison de telles fonctions, il est très redondant de laisser varier indépendamment les trois paramètres t_0 , ω_0 et Δt . Deux approches différentes ont prévalu [63]:

- la première approche est l'approche temps-fréquence, dans laquelle la largeur spatiale Δt des fonctions g est indépendante de la fréquence. Les fonctions s'écrivent sous la forme

$$g_{t_0, \omega_0}(t) = e^{i\omega_0 t} g_0(t - t_0)$$

où $g_0(t) = A_0 e^{-t^2/2\Delta t^2}$ Ce mode d'approximation est également appelé analyse de Fourier à fenêtre.

➤ une deuxième approche est l'approche *temps-échelle*, dans laquelle la largeur spatiale des fonctions g est inversement proportionnelle à la fréquence (le produit $w_0 \Delta t$ est constant et égal à c). On obtient alors, à un facteur de module 1 près, une forme relativement simple pour les différentes ondelettes :

$$g_{t_0, \Delta t}(t) = \frac{1}{\sqrt{\Delta t}} g_0 \left(\frac{t - t_0}{\Delta t} \right)$$

Où $g_0(t) = A_0 e^{-t^2/2\Delta t^2} e^{ict}$.

Les pavages du plan temps-fréquence donnent une idée de la différence des deux approches.

Dans le cas de l'analyse temps-fréquence, le pavage obtenu est un pavage par des domaines rectangulaires qui se déduisent les uns des autres par translation dans le plan temps-fréquence. Dans le cas de l'analyse temps-échelle, les domaines sont également de surface constante, mais ont une résolution fréquentielle relative $\Delta w/w_0$ constante.

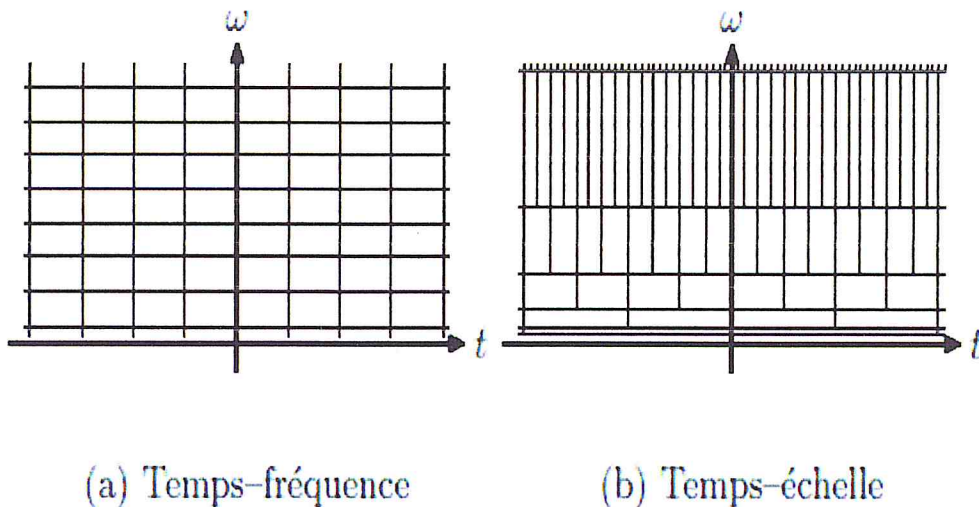


Figure. 3-2 : Pavages du plan temps-fréquence pour la représentation temps-fréquence et la représentation temps-échelle

Les ondelettes que nous allons utiliser dans la suite correspondent à l'analyse temps-échelle. Les avantages de ce mode de décomposition sont multiples. On dispose de moyens efficaces pour construire des bases discrètes pour lesquelles les calculs de transformation sont très rapides. L'analyse multi-échelles repose sur une forme de fonction unique. Enfin, dans l'analyse temps-échelle, les fonctions de base ont une taille de support proportionnelle à la résolution spatiale liée à leur fréquence par l'inégalité de Heisenberg.

III.1.3 Transformée en ondelettes continue

On choisit une fonction de base Ψ appelée ondelette qui vérifie la condition d'admissibilité [64]:

$$C_\psi = \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty \quad (3.1)$$

On remarque que si transformée de Fourier de Ψ est régulière, elle doit nécessairement s'annuler en 0, ce qui exclut d'office la fonction de Gabor que nous avons indiquée ci-dessus. En pratique, on considérera que la fonction de Gabor satisfait presque la condition d'admissibilité, dans la mesure où sa transformée de Fourier prend une valeur très petite (mais non nulle) en 0.

On définit la transformée en ondelettes continue Wf de la fonction f par la formule suivante :

$$Wf(t,s) = \int_{\mathbb{R}} f(\tau) \frac{1}{\sqrt{s}} \overline{\psi\left(\frac{\tau-t}{s}\right)} d\tau \quad (3.2)$$

La transformée inverse s'écrit

$$f(t) = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} W(\tau,s) \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) d\tau \frac{ds}{s^2} \quad (3.3)$$

De même que dans le cas de la transformée de Fourier, la transformation est quasi isométrique:

$$\|f\|_{L_2(\mathbb{R})}^2 = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} |Wf(t,s)|^2 dt \frac{ds}{s^2}. \quad (3.4)$$

En revanche, la représentation d'une fonction f par sa transformée en ondelettes n'est pas biunivoque, mais redondante, car une transformée en ondelettes vérifie une équation aux noyaux reproduisant :

$$Wf(t,s) = \frac{1}{C_\psi} \iint_{\mathbb{R}^2} W(\tau,\sigma) K(\tau,t,\sigma,s) dt \frac{ds}{s^2}$$

avec un noyau K défini par

$$K(\tau,t,\sigma,s) = \int_{\mathbb{R}} \frac{1}{\sqrt{\sigma s}} \psi\left(\frac{t'-t}{s}\right) \overline{\psi\left(\frac{t'-\tau}{\sigma}\right)} dt'$$

III.1.4 La transformée en ondelettes discrète

Morlet a proposé de construire des bases ou des frames de fonctions construits sur le modèle suivant [62]:

$$g_{t_0, \Delta t}(t) = \frac{1}{\sqrt{\Delta t}} g\left(\frac{t - t_0}{\Delta t}\right)$$

Où les valeurs possibles de Δt sont pris sur une échelle géométrique et les paramètres de translation sont proportionnels à Δt :

$$\begin{aligned} \Delta t &= b^j \\ t_0 &= k\Delta t \end{aligned}$$

Une gamme d'échelles Δt couramment utilisée est la gamme des échelles dyadiques 2^j , on obtient des familles constituées de fonctions de la forme $g_0(2^j(t - 2^{-j}k)) = g(2^j t - k)$ où j et k sont des entiers relatifs. La normalisation la plus couramment utilisée étant une normalisation en norme L2, on obtient des familles de fonctions $(\psi_{jk})_{j,k \in \mathbb{Z}}$ où $\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$.

Dans un article écrit en 1987, Stéphane Mallat [60] a tracé un parallèle entre les fonctions de représentation temps-échelle inspirées par le travail de Morlet et les filtres miroirs en quadrature étudiés par Burt, Adelson et Simoncelli pour effectuer de la compression d'images.

Il a mis en avant une certaine catégorie de décompositions en ondelettes qui peuvent être réalisées numériquement en un temps très court par une « transformée en ondelettes rapide », pour laquelle l'ondelette de base doit être écrite comme convolution infinie de filtres discrets. Plus exactement, soit un couple de filtres discrets (m_0, m_1) :

$$k \mapsto m_0[k] \quad k \in \mathbb{Z}, \quad (3.5)$$

$$k \mapsto m_1[k] \quad k \in \mathbb{Z}; \quad (3.6)$$

dont les transformées de Fourier $\omega \mapsto m_0(\omega)$ et $\omega \mapsto m_1(\omega)$ sont des fonctions 2π périodiques. On suppose qu'il existe une fonction d'échelle Φ et une ondelette Ψ de $L^2(\mathbb{R})$ telles que

$$\hat{\phi}(\omega) = \prod_{k=1}^{+\infty} m_0\left(\frac{\omega}{2^k}\right) \quad (3.7)$$

$$\hat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.8)$$

Sous certaines conditions sur m_0 et m_1 , la famille (ψ_{jk}) est une base orthogonale, et la décomposition en ondelettes d'une fonction échantillonnée peut être effectuée par un algorithme rapide constitué d'une cascade de filtrages et de sous-échantillonnages.

Cette approche réduit considérablement la complexité de la construction d'une ondelette. Au lieu de choisir une fonction, on choisit l'ensemble discret (et en général fini) des coefficients des deux filtres.

III.1.5 Analyses multi-résolutions

Le cadre d'analyse de ces ondelettes qui s'expriment à l'aide de filtres discrets s'est considérablement développé ces dernières années ; nous disposons d'un ensemble de théorèmes reliant les propriétés des ondelettes et celles de filtres discrets. Par ailleurs, il existe plusieurs familles classiques d'ondelettes qui portent en général soit le nom de leur créateur, soit celui d'une propriété.

III.1.5.1 Cadre théorique

Le cadre théorique posé par Stéphane Mallat est basé sur la notion d'analyse multi-résolutions. Une analyse multi-résolutions est une famille de sous-espaces de $L_2(\mathbb{R})$ notée $(V_j)_{j \in \mathbb{Z}}$, qui ont les propriétés suivantes :

$$V_j = \left\{ \sum_{k \in \mathbb{Z}} a_k \phi_{jk} : a_k \in \mathbb{R} \right\} \quad (\text{Espace de RIESZ}) \quad (3.9 \text{ a})$$

$$V_j \subset V_{j+1} \quad (3.9 \text{ b})$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\} \quad (3.9 \text{ c})$$

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L_2(\mathbb{R}) \quad (3.9 \text{ d})$$

Nous pouvons faire les remarques suivantes :

- L'hypothèse (3.9 a) signifie que V_j est un espace de Riesz engendré par la famille $(\phi_{jk})_{k \in \mathbb{Z}}$. Sa définition dépend de la topologie choisie pour l'espace fonctionnel. On peut le définir plus rigoureusement comme l'adhérence de l'espace de combinaisons linéaires finies de fonctions ϕ_{jk} . Cette propriété impose une contrainte sur la fonction ϕ . Pour une topologie L_2 , la correspondance doit être continue. Une fonction ϕ avec une décroissance en espace trop faible n'est donc pas admissible.

$$\begin{aligned} \ell_2(\mathbb{Z}) &\rightarrow L_2(\mathbb{R}) \\ (a_k)_{k \in \mathbb{Z}} &\mapsto \sum_{k \in \mathbb{Z}} a_k \phi_{0k} \end{aligned}$$

- Intuitivement, nous pouvons considérer que l'ensemble des fonctions de V_{j+1} constitue un ensemble plus « riche » ou plus « dense » que V_j , ce qui ne signifie nullement de relation d'inclusion. L'hypothèse (3.9b) l'impose.

Pour des raisons d'invariance par translation et par changement d'échelles relatives, on peut vérifier que cette hypothèse est équivalente à supposer que $\Phi \in V_1$, ce qui signifie qu'il existe une suite de coefficients $(m_0[k])_{k \in \mathbb{Z}}$ telle que

$$\phi(t) = 2 \sum_{k \in \mathbb{Z}} m_0[k] \phi(2t - k) \quad (3.10)$$

C'est ainsi que nous voyons apparaître le filtre discret m_0 .

- L'hypothèse (3.9c) est formulée plus pour des raisons de principe, car elle est toujours vérifiée. L'hypothèse (3.9d) l'est pourvu que la fonction Φ' ne soit pas nulle en $w = 0$.

Les ondelettes apparaissent naturellement comme un moyen d'écrire la différence entre deux espaces V_j et V_{j+1} consécutifs. On construit pour cela un espace de Riesz W_0 tel que :

$$V_0 \oplus W_0 = V_1 \quad (3.11)$$

L'espace W_0 est engendré par une fonction Ψ :

$$W_0 = \left\{ t \mapsto \sum_{k \in \mathbb{R}} d_k \psi(t - k) : d_k \in \mathbb{R} \right\}$$

Ceci impose que la fonction soit dans l'espace V_1 , et s'écrive comme combinaison linéaire des fonctions $(t \mapsto \phi(2t - k))_{k \in \mathbb{Z}}$:

$$\psi(t) = \sum_{k \in \mathbb{Z}} m_1[k] \phi(2t - k) \quad (3.12)$$

Ainsi apparaît le deuxième filtre discret m_1 .

On montre que les fonctions Φ et Ψ sont alors définies par les seules données des *filtres* discrets m_0 et m_1 . Les formules sont

$$\hat{\phi}(\omega) = \prod_{k=1}^{+\infty} m_0\left(\frac{\omega}{2^k}\right) \quad \text{par itération de (3.10)} \quad (3.13 \text{ a})$$

$$\hat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \quad \text{par (3.12)} \quad (3.13 \text{ b})$$

III.1.5.2 Les bases d'ondelettes

La relation (3.11) se transpose à toutes les échelles j :

$$V_j \oplus W_j = V_{j+1} \quad (3.14)$$

et on obtient par itérations de cette relation :

$$V_j \oplus W_j \oplus \dots \oplus W_{j'-1} = W_{j'} \quad \text{si } j < j' \quad (3.15)$$

En faisant tendre j_0 vers $+1$ (et éventuellement j vers $-\infty$), on obtient deux relations :

$$L_2(\mathbb{R}) = V_j \oplus \overline{\bigoplus_{j'=j}^{+\infty} W_{j'}} \quad \text{quel que soit } j \in \mathbb{Z} \quad (3.16)$$

$$L_2(\mathbb{R}) = \overline{\bigoplus_{j'=-\infty}^{+\infty} W_{j'}} \quad (3.17)$$

La réunion des bases de Riesz de chacun des espaces de ces sommes directes fournit ainsi plusieurs bases d'ondelettes :

$$\mathcal{B}_j = \{\phi_{jk} : k \in \mathbb{Z}\} \cup \{\psi_{j'k} : j' \geq j, k \in \mathbb{Z}\} \quad (3.18)$$

$$\mathcal{B} = \{\psi_{jk} : j \in \mathbb{Z}, k \in \mathbb{Z}\} \quad (3.19)$$

III.1.5.3 Transformée en ondelettes

La transformée en ondelettes s'applique à des signaux échantillonnés sur une grille discrète, et en général cet échantillonnage consiste à approcher une fonction de $L_2(\mathbb{R})$ par

$$f = \sum_{k \in \mathbb{Z}} 2^{j/2} f[k/2^j] \phi_{jk}$$

où l'échantillon $f[k/2^j]$ peut lui aussi être estimé par :

$$f[k/2^j] \simeq f(k/2^j)$$

Le signal dont on part est donc représenté sur une base de Riesz de V_j . Appliquer à ce signal une transformation en ondelettes jusqu'à l'échelle $L \in \mathbb{Z}$ revient à représenter ce signal sur une base adaptée à la somme directe :

$$V_L \oplus W_L \oplus W_{L+1} \oplus \dots \oplus W_{j-1}$$

L'algorithme de transformation est itératif et consiste à remplacer la représentation d'une composante sur V_j par une représentation sur $V_{j-1} \oplus W_{j-1}$. On passe ainsi successivement par des décompositions adaptées aux différentes sommes directes suivantes :

$$\begin{aligned} & V_{j-1} \oplus W_{j-1} \\ & V_{j-2} \oplus W_{j-2} \oplus W_{j-1} \\ & \vdots \\ & V_L \oplus W_L \oplus W_{L+1} \oplus \dots \oplus W_{j-1} \end{aligned}$$

III.1.5.4 Filtrés duaux, ondelettes duales

L'itération de la base de transformation en ondelettes est donc un changement de base entre les deux décompositions suivantes :

$$V_{j+1} \rightarrow V_j \oplus W_j$$

Elle s'écrit comme une correspondance

$$\begin{aligned} \ell_2(\mathbb{Z}) &\rightarrow \ell_2(\mathbb{Z}) \times \ell_2(\mathbb{Z}) \\ (a_{j+1,k})_{k \in \mathbb{Z}} &\mapsto [(a_{jk})_{k \in \mathbb{Z}}, (d_{jk})_{k \in \mathbb{Z}}] \end{aligned}$$

Si on note A_j et D_j les fonctions 2π périodiques dont les coefficients sont les suites discrètes $k \mapsto a_{jk}$ et $k \mapsto d_{jk}$:

$$\begin{aligned} A_j(\omega) &= \sum_{k \in \mathbb{Z}} a_{jk} e^{-ik\omega} \\ D_j(\omega) &= \sum_{k \in \mathbb{Z}} d_{jk} e^{-ik\omega} \end{aligned}$$

l'itération de base s'écrit comme l'application d'une matrice de transfert :

$$\begin{bmatrix} A_j(2\omega) \\ D_j(2\omega) \end{bmatrix} = \begin{bmatrix} m_0(\omega) & m_0(\omega + \pi) \\ m_1(\omega) & m_1(\omega + \pi) \end{bmatrix} \begin{bmatrix} A_{j+1}(\omega) \\ A_{j+1}(\omega + \pi) \end{bmatrix}$$

Une condition nécessaire pour que cette transformation soit inversible est donc que la matrice de transfert

$$T(\omega) = \begin{bmatrix} m_0(\omega) & m_0(\omega + \pi) \\ m_1(\omega) & m_1(\omega + \pi) \end{bmatrix}$$

soit bornée d'inverse bornée sur $[0, 2\pi]$. Dans ce cas, on appelle matrice de transfert dual la matrice $\tilde{T}(\omega) = T(\omega)^{-T}$. Il existe deux autres fonctions 2π -périodiques \tilde{m}_0 et \tilde{m}_1 telles que $\tilde{T}(\omega)$ s'écrit :

$$\tilde{T}(\omega) = \begin{bmatrix} \tilde{m}_0(\omega) & \tilde{m}_0(\omega + \pi) \\ \tilde{m}_1(\omega) & \tilde{m}_1(\omega + \pi) \end{bmatrix}$$

Ces filtres définissent les ondelettes duales $\hat{\phi}$ et $\hat{\psi}$ par des relations identiques à (3.13a) et (3.13b) :

$$\hat{\phi}(\omega) = \prod_{k=1}^{+\infty} \tilde{m}_0\left(\frac{\omega}{2^k}\right) \quad (3.20 \text{ a})$$

$$\hat{\psi}(\omega) = \tilde{m}_1\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.20 \text{ b})$$

Les fonctions $\tilde{\phi}$ et $\tilde{\psi}$ sont donc des ondelettes duales dans le sens où pour tout j , on a les formules de décomposition sur $L^2(\mathbb{R})$:

$$f = \sum_{k \in \mathbb{Z}} \langle f, \tilde{\phi}_{jk} \rangle \phi_{jk} + \sum_{j' \geq j, k \in \mathbb{Z}} \langle f, \tilde{\psi}_{j'k} \rangle \psi_{j'k} \quad (3.21)$$

pour tous $j \in \mathbb{Z}$ et $f \in L^2(\mathbb{Z})$, et en faisant tendre j vers $-\infty$ la formule de décomposition homogène :

$$f = \sum_{j, k \in \mathbb{Z}} \langle f, \tilde{\psi}_{jk} \rangle \psi_{jk} \quad (3.22)$$

III.1.5.5 L'algorithme de la transformée en ondelettes rapide

Les coefficients des filtres m_0 , m_1 et des filtres duaux \tilde{m}_0 et \tilde{m}_1 interviennent dans le calcul des changements de base

$$\{\phi_{jk} : k \in \mathbb{Z}\} \cup \{\psi_{jk} : k \in \mathbb{Z}\} \rightarrow \{\phi_{j+1,k} : k \in \mathbb{Z}\}$$

avec les formules suivantes :

- dans le sens de transformation dit « direct » (forward wavelet transform), on a :

$$a_{jk} = 2 \sum_{l \in \mathbb{Z}} \tilde{m}_0[k] a_{j+1, 2l-k}$$

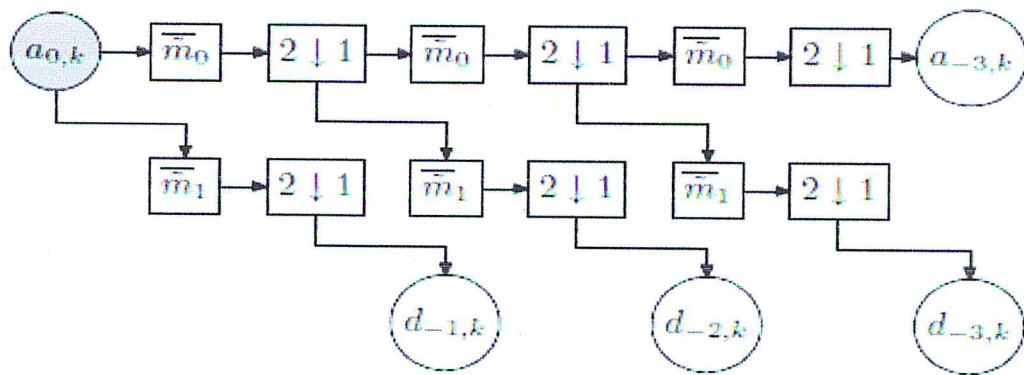
$$d_{jk} = 2 \sum_{l \in \mathbb{Z}} \tilde{m}_1[k] a_{j+1, 2l-k}$$

- et dans le sens de transformation inverse (inverse wavelet transform), on obtient

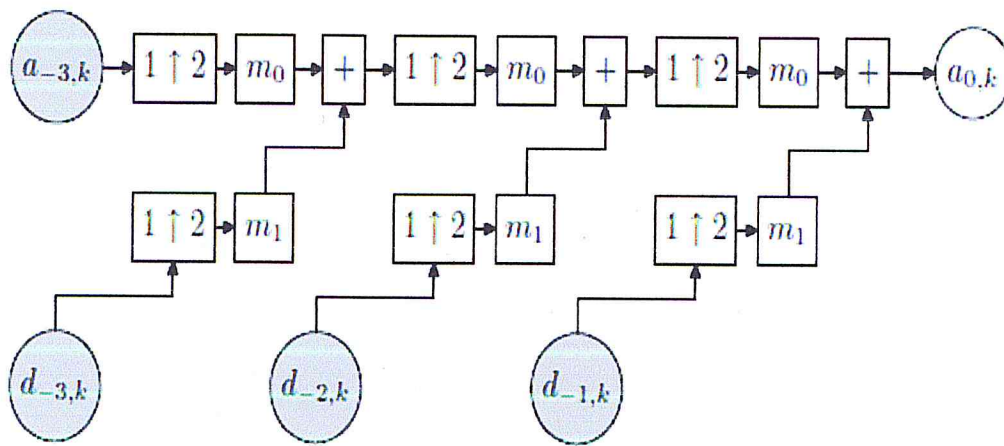
$$a_{j+1,k} = \frac{1}{2} \sum_{l \in \mathbb{Z}} m_0[2l-k] a_{jl} + m_1[2l-k] d_{jl}$$

Des schémas de transformation directe et de transformation inverse sont représentés sur la figure 3.3 entre les échelles $j = 0$ et $j = -3$.

Pour un nombre fini N d'échantillons, une transformation en ondelettes (jusqu'à n'importe quelle profondeur autorisée par la taille de l'échantillon) prend moins de $A \times N$ opérations, où la constante A dépend naturellement de la taille des filtres. Ceci est en théorie meilleur qu'une transformée de Fourier rapide qui prend de l'ordre de $N \log N$ opérations.



(a) Transformée directe



(b) Transformée inverse

Figure. 3-3 : Transformées en ondelettes rapides. Les cercles sur fond grisé désignent les composantes d'entrée, tandis que les cercles sur fond blanc désignent les composantes de sortie

III.1.5.6 Les ondelettes orthogonales

Les ondelettes orthogonales sont des ondelettes ψ telles que la famille $(t \mapsto 2^{j/2} \psi(2^j t - k))_{j,k \in \mathbb{Z}}$ soit une base orthogonale de $L^2(\mathbb{R})$. C'est le cas dès que $\phi = \bar{\phi}$ et $\psi = \bar{\psi}$, ce qui équivaut à écrire que la matrice de transfert et la matrice de transfert duale sont égales, soit encore que la matrice de transfert est unitaire pour tout w . Ceci se traduit par la contrainte sur les filtres m_0 et m_1 par :

$$\begin{aligned} |m_0(\omega)|^2 &+ |m_0(\omega + \pi)|^2 &= 1 & \forall \omega \\ m_0(\omega) \overline{m_1(\omega)} &+ m_0(\omega + \pi) \overline{m_1(\omega + \pi)} &= 0 & \forall \omega \\ |m_1(\omega)|^2 &+ |m_1(\omega + \pi)|^2 &= 1 & \forall \omega \end{aligned}$$

Dans ce cas, les filtres m_0 et m_1 sont appelés filtres miroirs en quadrature, selon la terminologie d'Esteban et Galand et reprise par Adelson et Simoncelli. De plus les sommes directes qui apparaissent dans les formules ci-dessus sont toutes orthogonales. En pratique le choix des filtres se réduit au seul choix de m_0 , car alors un choix pour m_1 s'impose :

$$m_1(\omega) = e^{i\omega} \overline{m_0(\omega + \pi)}$$

Historiquement, les premières ondelettes qui ont été mises au point sont les ondelettes orthogonales (Meyer, Mallat), si bien que les familles d'ondelettes non orthogonales ont reçu l'appellation d'ondelettes biorthogonales. Le préfixe « bi » est censé rappeler que deux bases d'ondelettes sont utilisées, une pour l'analyse (la base duale) et une pour la reconstruction. Une étude systématique des ondelettes biorthogonales a été menée par Cohen, Daubechies et al [63].

Il existe un certain nombre de familles d'ondelettes orthogonales couramment utilisées. Les plus connues sont sans doute les ondelettes de Daubechies. Ces ondelettes résultent de compromis optimaux entre deux critères contradictoires : le nombre de moments nuls des ondelettes et la taille de leur support (ces deux critères contradictoires rappellent dans une certaine mesure l'inégalité de Heisenberg). Il existe d'autres familles d'ondelettes orthogonales, comme les *coiflets*, du nom de Ronald Coifman, ou les *symmlets* qui sont des ondelettes presque symétriques [61].

Les bases d'ondelettes orthonormées ont un avantage théorique considérable dans les problèmes de compression ou de débruitage : la métrique d'erreur utilisée est en général la métrique L2, et celle-ci s'exprime très simplement avec les coefficients d'une décomposition dans une base orthonormale. Dans le cas du débruitage, il se trouve également qu'un bruit blanc gaussien a une décomposition également très simple dans une base orthonormale : les coefficients sont alors également des variables gaussiennes indépendantes centrées et de même variance.

En pratique, en revanche, les ondelettes orthogonales n'offrent pas la même souplesse dans leur conception que les ondelettes biorthogonales, et on peut également montrer qu'elles ne peuvent jamais être symétriques (à un cas trivial près). Il faut noter que la relation duale entre la base d'analyse des $\tilde{\psi}_{jk}$ et de reconstruction des ψ_{jk} est symétrique, et que les rôles peuvent être interchangés, on peut très bien utiliser la base $(\psi_{jk})_{j,k \in \mathbb{Z}}$ pour l'analyse et $(\tilde{\psi}_{jk})_{j,k \in \mathbb{Z}}$ pour la reconstruction, pour avoir la formule :

$$f = \sum_{j,k \in \mathbb{Z}} \langle \psi_{jk}, f \rangle \tilde{\psi}_{jk}$$

que l'on peut comparer à la formule (3.22).

III.1.6 Décroissance des coefficients, régularité et approximation

Pour que la convergence d'une décomposition d'une fonction sur une base d'ondelettes soit rapide, il faut que les coefficients d'ondelettes décroissent rapidement quand $j \rightarrow +\infty$.

Cette décroissance est liée au nombre de moments nuls de l'ondelette duale $\tilde{\psi}$.

On dit que $\tilde{\psi}$ a p moments nuls si [64]:

$$\int_{\mathbb{R}} \tilde{\phi}(t) t^k dt = 0$$

pour tout k dans $\{0, \dots, p-1\}$. Ceci revient à écrire que la transformée de Fourier de $\tilde{\psi}$ a un zéro d'ordre p en $w = 0$, ou à dire que $\tilde{\psi}$ est orthogonale à tous les polynômes de degré inférieur à p.

On peut montrer alors que si une fonction f est p fois continûment dérivable dans un intervalle I , ses coefficients d'ondelettes $\langle \tilde{\psi}_{jk}, f \rangle$ décroissent en $2^{-j(p+1/2)}$ dans I , c'est-à-dire qu'il existe une borne M telle que :

$$\left| \langle \tilde{\psi}_{jk}, f \rangle \right| \leq M 2^{-(p+1/2)j} \quad \text{si } \text{supp } \tilde{\psi}_{jk} \subset I$$

Pour montrer cette majoration, il suffit d'appliquer la formule de Taylor à la fonction f autour du centre de l'ondelette $\tilde{\psi}_{jk}$. Si $u = k/2^j$, on a :

$$f(t) = \sum_{k=0}^{p-1} \frac{(t-u)^k}{k!} f^{(k)}(u) + (t-u)^p r(t)$$

où la fonction $r(t)$ est bornée par la dérivée p ième de f . Quand on fait le produit scalaire de f avec $\tilde{\psi}_{jk}$, la somme de termes polynômiaux disparaît, et il ne reste que :

$$\begin{aligned} \langle \tilde{\psi}_{jk}, f \rangle &= \int (t-u)^p r(t) \overline{\tilde{\psi}_{jk}(t)} dt \\ &= \int t^p r(t+u) \overline{\tilde{\psi}_{j0}(t)} dt \end{aligned}$$

dont on montre par un simple changement de variable qu'il est majoré par $M 2^{-(p+1/2)j}$. On voit donc que la régularité locale de la fonction conditionne la décroissance en échelles des coefficients d'ondelettes. Une quasi-réciproque est également vraie : si la décroissance

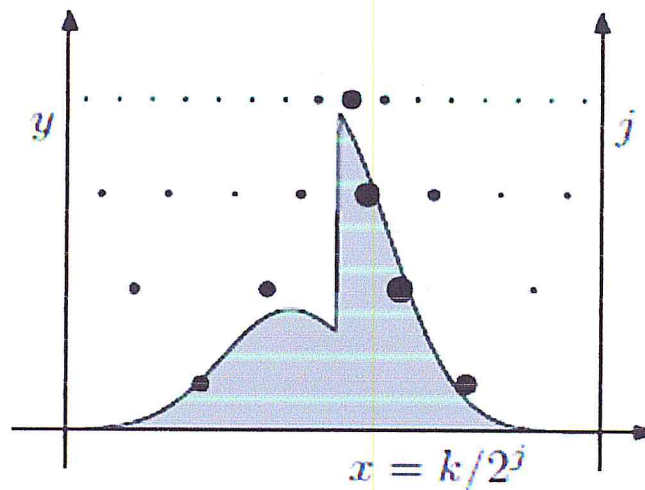


Figure. 3-4 : Coefficients d'ondelettes d'une fonction à support compact. On voit notamment que les coefficients restent importants autour des singularités pour des j croissants.

des coefficients est de l'ordre $M 2^{-(p+1/2)j}$, et si ψ est p -Lipschitz, alors la fonction est r -Lipschitz pour tout $r < p$.

Pour illustrer le lien entre coefficients et régularité, la décomposition en ondelettes d'une fonction à support compact avec une singularité est donnée par la figure 3.4.

III.2 Le format FBI – Wavelet Scalar Quantization (WSQ)

WSQ est une norme de compression d'images d'empreintes digitales, permettant de représenter ces dernières par un format d'image compressée c'est-à-dire elle permet de préciser le format d'échange des images compressées. L'objectif de la norme est de compléter la performance du standard JPEG.

III.2.1 Un aperçu général sur l'algorithme WSQ

La compression d'images par l'algorithme (WSQ) s'effectue en trois étapes principales selon le processus suivant (figure 3.5) :

- transformation de l'image de l'empreinte en ondelettes,
- quantification scalaire uniforme des coefficients,
- codage entropique sans perte des indices quantifiés.

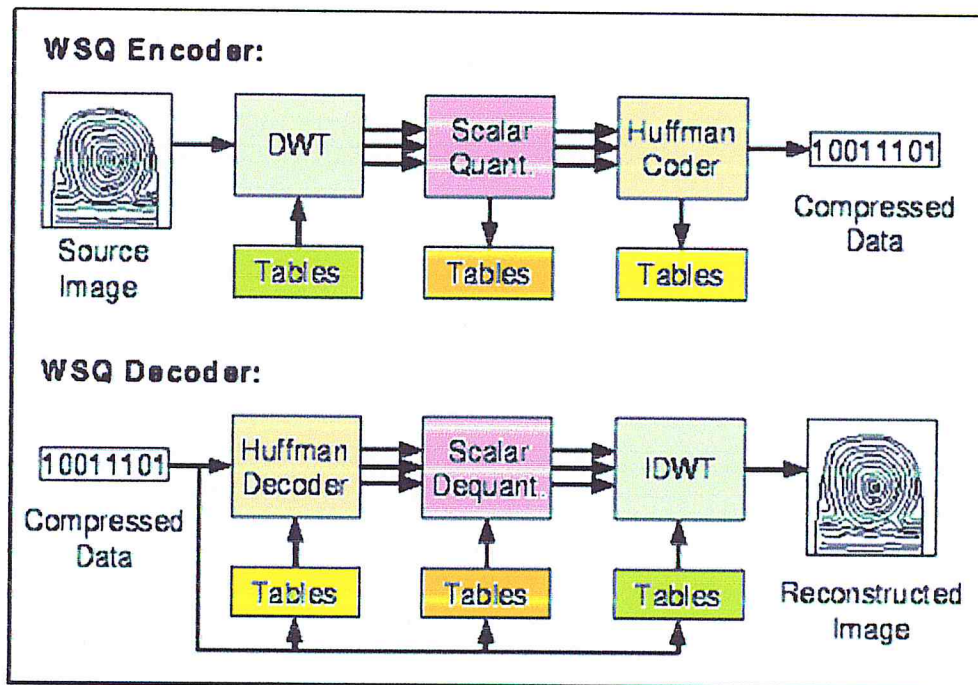


Figure 3-5 : Les principaux processus de l'encodeur et le décodeur WSQ

La norme définit une classe de codeurs et un seul décodeur général suffisant pour décoder les données d'image compressées par n'importe quel encodeur compatible. Dans un codeur WSQ, la source de l'image numérisée est décomposée en 64 sous-images à deux dimensions symmetric wavelet transform (SWT) avec des coefficients d'ondelettes ayant des valeurs de type réel [55].

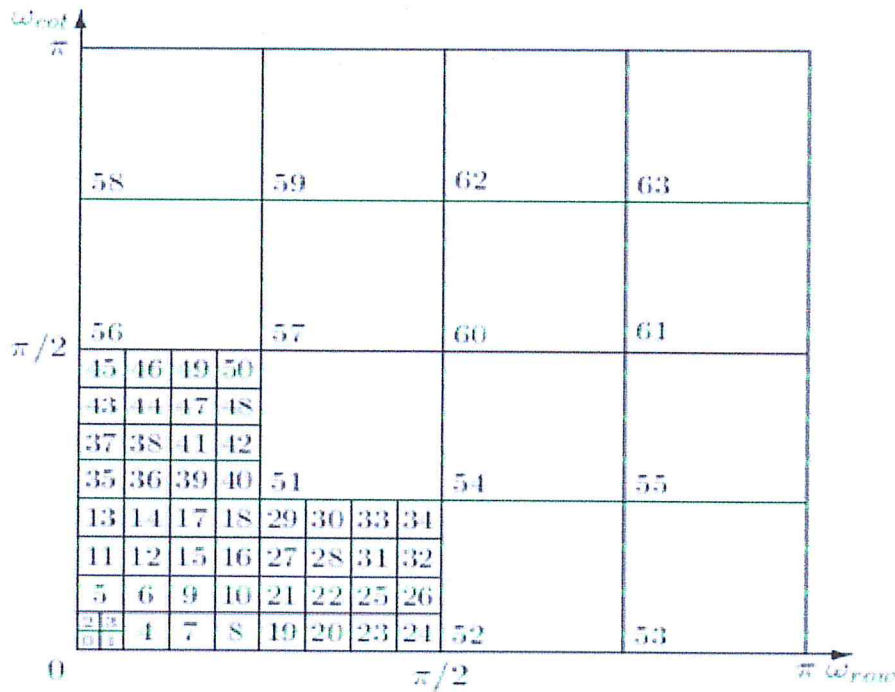


Figure. 3-6 : Décomposition en sous-images selon le format WSQ

Les sous-images de la transformation en ondelettes symétriques sont ensuite transmises à une banque d'indices scalaires quantifiés de façon uniforme pour être codées par l'utilisation de deux types de codage entropique : codage run-length et codage Huffman. La compression de données d'image utilise trois types de tableaux : un tableau pour les spécifications de la transformation en ondelettes, un tableau pour les quantifications scalaires et un tableau pour le codage.

Il existe deux principaux formats spécifiés pour la compression des données : un format d'échange de toutes les tables nécessaires au décodage d'une image après la transmission entre les applications, et un abrégé pour l'utilisation au sein d'une seule application dans laquelle les données sous forme de tableaux proviennent d'autres sources. La syntaxe de données compressées est étroitement calquée sur la syntaxe employée dans le reste de la compression d'image JPEG standard [57]. Des marqueurs de deux-octets sont inclus pour permettre au décodeur d'analyser le flux de bits compressé et de localiser des informations secondaires telles que le tableau des spécifications avant le décodage des données de l'image compressée. Le décodeur WSQ analyse des données compressées et les extraits des tableaux nécessaires au processus de décodage.

Un décodeur entropique utilise les tables de Huffman pour décoder les sous-images compressées et les indices de quantification scalaire sont ensuite décodés pour reconstruire les coefficients d'ondelettes, qui sont des approximations des coefficients d'ondelettes d'origine. Les coefficients passent alors par la transformation en ondelettes inversée pour produire l'image reconstruite. La norme permet aux codeurs d'utiliser des filtres d'ondelettes provenant de l'une des deux classes de la banque de filtres pour reconstruire l'image de façon linéaire parfaite, et ce en liaison avec deux algorithmes de transformation en ondelettes et des spécifications pour la quantification scalaire et pour le codage entropique. Le CJIS (les

services d'information de la division criminelle du FBI) spécification inclut également des tests de conformité pour les encodeurs, décodeurs, et le format des données compressées.

La partie «codage» utilise tour à tour une Transformation en Ondelettes Discrètes qui décompose l'image en composantes (64 sous-images) associées aux différentes sous-bandes, une *quantification* pour chaque sous-image, et un *codage entropique*, sans pertes. La partie «reconstruction» applique successivement les opérations inverses de décodage, de "quantification inverse" qui consiste à ramener les pixels numérisés dans l'intervalle d'amplitude initiale, et la Transformation en Ondelettes Discrètes inverse, qui reconstruit l'image à partir des sous-images quantifiées.

III.2.1.1 La transformée en ondelettes

Une banque de filtres de reconstruction unidimensionnelle à deux canaux est décrite dans la Figure (3-7).

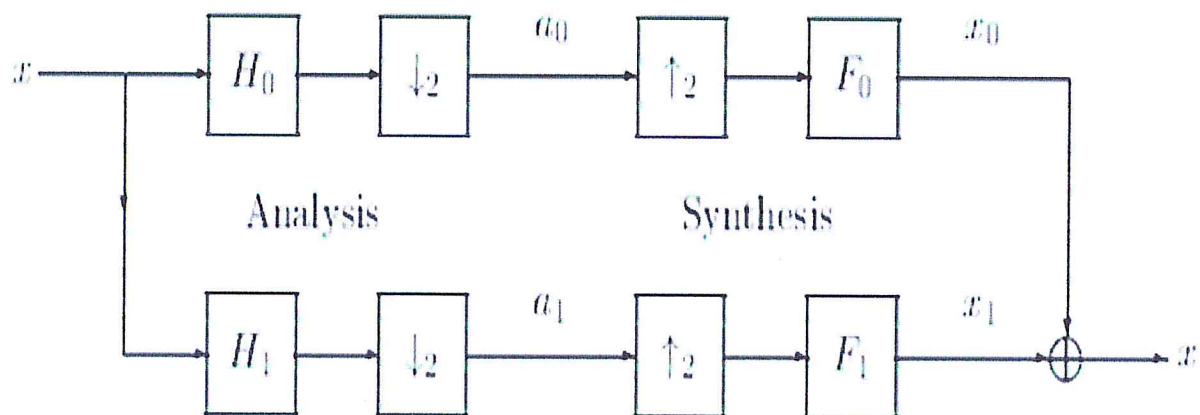


Figure. 3-7 : Un banc de filtres de reconstruction unidimensionnelle idéale

Pour les applications de traitement d'images, il est nécessaire de préciser la manière dont les bords d'image doivent être limités à la durée d'entrée d'un signal X comme un vecteur ligne ou colonne à partir d'une image numérisée. En général, une seule banque de filtres est capable de fournir un certain nombre de transformations différentes selon la manière dont elle est appliquée par les entrants à durée limitée. La façon la plus simple de gérer une entrée de longueur N_0 est d'appliquer des filtres à réponse impulsionnelle finie par N_0 convolutions périodiques, suivie de 2:1 à l'échantillonnage circulaire [55].

Il ya deux problèmes avec cette approche: d'abord, la périodisation introduit un saut discontinu dans le signal, ce qui entraîne une plus grande variation des coefficients passe haut des sous-images, et, deuxièmement, l'échantillonnage à 2 :1 n'est possible que lorsque N_0 est pair.

De plus, la variance des hautes bandes affecte la qualité de la performance de la quantification, et la norme n'impose aucune contrainte sur les dimensions de l'image (par exemple, les dimensions ne doivent pas être de puissance 2, ou même divisibles par 2), de sorte que la norme WSQ traite ces deux problèmes à la fois par l'application de la banque de filtres à une extension périodique symétrique. Une transformation de cette manière est appelée une transformation en ondelettes symétriques (SWT) [58].

Deux extensions symétriques sont présentées dans la figure 3.8. Notons que les deux ont la même longueur, indépendamment de savoir si N_0 est pair ou impair. Étant donné que la longueur (ou période) de l'entrée a été doublée par le processus d'extension symétrique, la question cruciale est de veiller à ce que la transformation ne dépasse pas la taille du signal transformé. Une transformation d'ondelette symétrique n'est pas expansive si le signal d'origine de la longueur N_0 peut être reconstruit de façon parfaite juste par la transformation des coefficients N_0 . Ceci est accompli en utilisant des filtres à phase linéaire conçus de façon à ce que les bas-échantillons des sous-images soient aussi symétriques, et peuvent donc être tronqués, sans perte d'information. La norme WSQ [59] définit deux catégories distinctes de la SWT pour les deux familles de banques de filtres non triviales pour la reconstruction parfaite à phase linéaire.

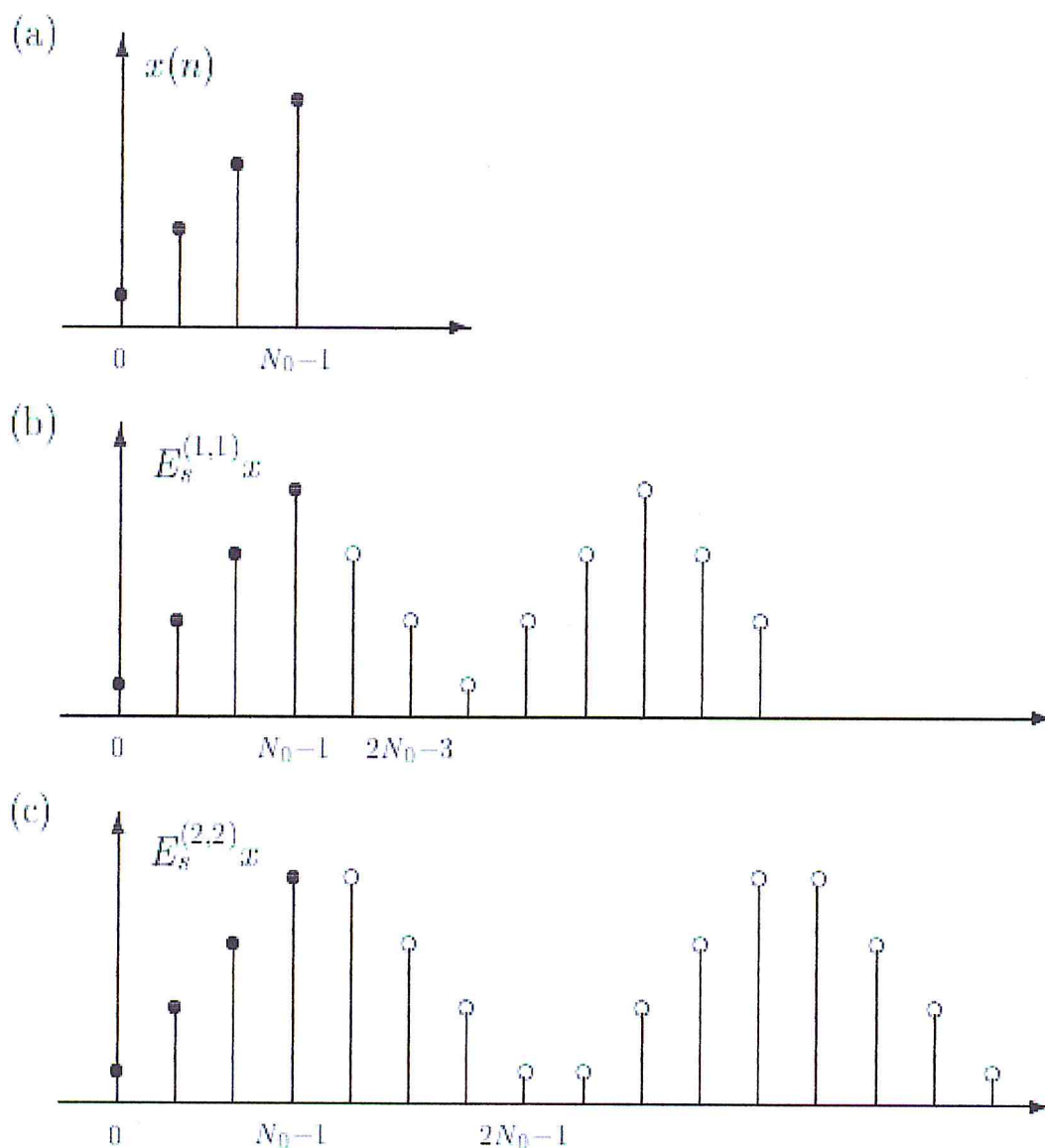


Figure. 3-8 : deux types de filtres à une extension périodique symétrique

L'extension $y = E(1,1)$, comme le montre la figure 3-8 (b), est utilisée avec des filtres à phase linéaire, et de longueurs impaires, tels qu'un filtre passe-bas (h_0) est symétrique par rapport au $n = 0$ et un filtre passe-haut (h_1) est symétrique au $n = -1$. L'extension $y = E(2,2)$ montrée dans la figure 3-8 (c) est utilisée avec des filtres à phase linéaire et de longueur paire contenant une phase de filtres passe-bas symétriques et une phase de filtres passe-haut antisymétrique, les deux centrées à $-1/2$. Avec ces conventions, lorsque N_0 est pair, il est nécessaire de transmettre simplement les coefficients $N_0/2$ dans les deux canaux passe-bas et passe-haut. Lorsque N_0 est impair, la reconstruction parfaite ne peut être assurée par la transmission des coefficients $(N_0 + 1)/2$ dans le canal passe-bas et des coefficients $(N_0 - 1)/2$ dans le canal passe-haut. Ainsi, le SWT spécifié par la norme WSQ est non expansif pour les signaux d'entrée de longueurs paire ou impaire, de sorte, il n'y a pas de contraintes quant aux dimensions de l'image, selon les exigences du FBI. Le codeur transmet les filtres d'analyse h_0 et h_1 , avec les données d'image compressées pour que le décodeur soit en mesure d'utiliser ces informations afin de construire les filtres de synthèse f_0 et f_1 .

Le type d'ondelette utilisée est l'ondelette biorthogonale 9/7 qui fait partie de la famille des ondelettes biorthogonales symétriques de Cohen, Daubechies et Feveau. Les filtres passe-bas associés aux ondelettes 9/7 possèdent ainsi $p - 9$ coefficients à l'analyse, $p - 7$ coefficients à la synthèse et sont décrits dans le Tab 3-1.

n	$h_0[n]$	$\tilde{h}_0[n]$
0	0.85269867900940	0.78848561640566
1	0.37740285561265	0.41809227322221
2	-0.11062440441842	-0.04068941760956
3	-0.02384946501938	-0.06453888262894
4	0.03782845550699	

TAB. 3-1 : Coefficients des réponses impulsionnelles symétriques des filtres passe-bas d'analyse $h_0[n]$ et de synthèse $\tilde{h}_0[n]$ associés à l'ondelette CDF 9/7.

Les ondelettes biorthogonales 9/7 sont illustrées par la figure 3.9 et possèdent $N = 4$ moments nuls à l'analyse et $N = 4$ à la synthèse. Elles possèdent un grand nombre de moments nuls pour un support relativement court. De plus, elles sont symétriques et très proches de l'orthogonalité. C'est une caractéristique importante en codage car elle assure un rapprochement important entre l'erreur de reconstruction et l'erreur de quantification, en terme d'erreur quadratique moyenne. Antonini et Barlaud furent les premiers à montrer la supériorité de la transformée en ondelettes biorthogonale 9/7 pour la décorrélation d'images naturelles [53]. A partir de là, elle a été très utilisée en codage d'images et notamment par le format WSQ.

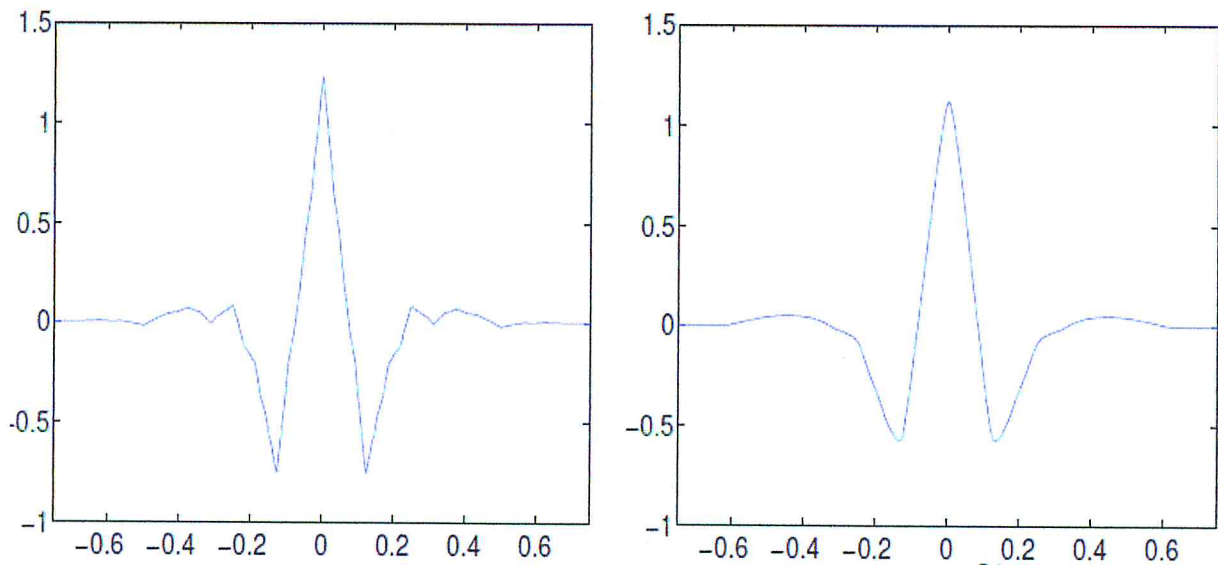


Figure. 3-9 : Ondelette CDF 9/7 d'analyse ψ et sa duale $\tilde{\psi}$.

Après la transformée, malgré une légère augmentation de la dynamique d'amplitude, les 64 sous-images présentent des variances fortement réduites, ce qui facilite leur compression par quantification et codage entropique.

III.2.1.2 La quantification

Le choix de WSQ s'est porté sur la quantification scalaire (pixel par pixel) uniforme. Son intérêt est qu'elle est facilement mise en œuvre. De plus, elle est complétée par un codeur entropique. Un autre avantage consiste à optimiser les pas non-uniformes des quantificateurs (ou les cellules de quantificateurs vectoriels) afin de minimiser la distorsion quadratique pour un signal donné. Mais ce procédé ne tient pas compte du codeur, et fait en partie le travail de celui-ci. L'association quantification uniforme + codage entropique reste par conséquent une référence.

D'autres améliorations sont possibles dans le contexte «uniforme» par :

- l'utilisation d'une « zone morte » pour les faibles valeurs du signal à quantifier et qui est déjà prise en considération;
- le cas vectoriel qui apporte un gain théorique relativement faible (< 0.8 dB) sur le rapport signal à bruit [55] pour des dimensions raisonnables (4, 8 ou 16), et dont l'intérêt pourrait surtout être l'utilisation de codeurs adaptés à la structure vectorielle. Cette solution n'a toutefois pas été adoptée pour les résultats présentés par WSQ en raison du choix simple du codeur entropique.

La quantification est effectuée ici indépendamment pour chaque sous-image (on ne tient pas compte des résidus de corrélation inter-bande). Elle introduit un bruit, dénommé bruit de quantification, qui va déterminer le niveau de distorsion obtenu après reconstruction de l'image.

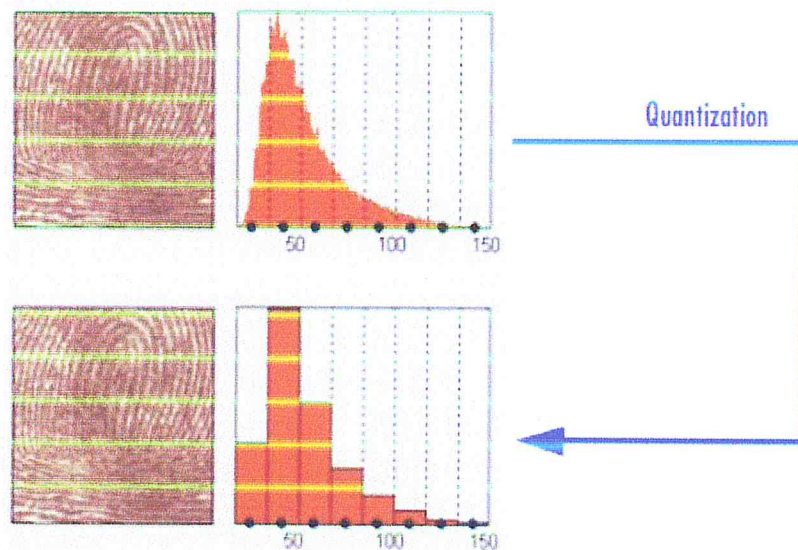


Figure. 3-10 : Quantification

III.2.1.3 Le codeur entropique

Le codeur entropique complète le travail des quantificateurs par une technique de compression sans pertes (effectuée indépendamment pour chaque sous-image). Il applique une technique de codage de Huffman global où chaque pixel est codé indépendamment de ses voisins, en fonction de la distribution statistique de la sous-image. Dans ce cas, il est légitime de négliger le débit nécessaire pour transmettre la table de codage. Ainsi trois critères de compression par le débit binaire ont pu être testés : débit binaire sans codeur (alloué aux quantificateurs), débit binaire obtenu par codage de Huffman, et calcul de l'entropie pixel par pixel [34].

III.2.2 Comparaison entre WSQ et JPEG 2000

On retrouve la transformation en ondelettes discrètes bidimensionnelles dans les deux schémas des normes WSQ et JPEG 2000 qui comportent, toutefois, des différences (figure 3.11):

- JPEG 2000 utilise le filtre de Daubechies (9/7) pour la compression avec perte et le filtre Daubechies (3/5) pour la compression sans perte, alors que WSQ utilise seulement le filtre de Daubechies (9/7).
- Concernant l'arbre de décomposition : JPEG 2000 applique l'arbre de décomposition dyadique basé sur le schéma de Mallat tandis que WSQ applique une structure fixe et un grand nombre de sous-bandes [50]. La décomposition en grand nombre sert à améliorer la compression par un rapprochement de l'orthnormalisation. Ainsi, la structure de décomposition va influencer sur la longueur du codage zéro alors que le JPEG2000 utilise le bit-plan de balayage afin de permettre un contrôle plus précis de la réalisation d'une compression arbitraire avec un taux spécifié [34].

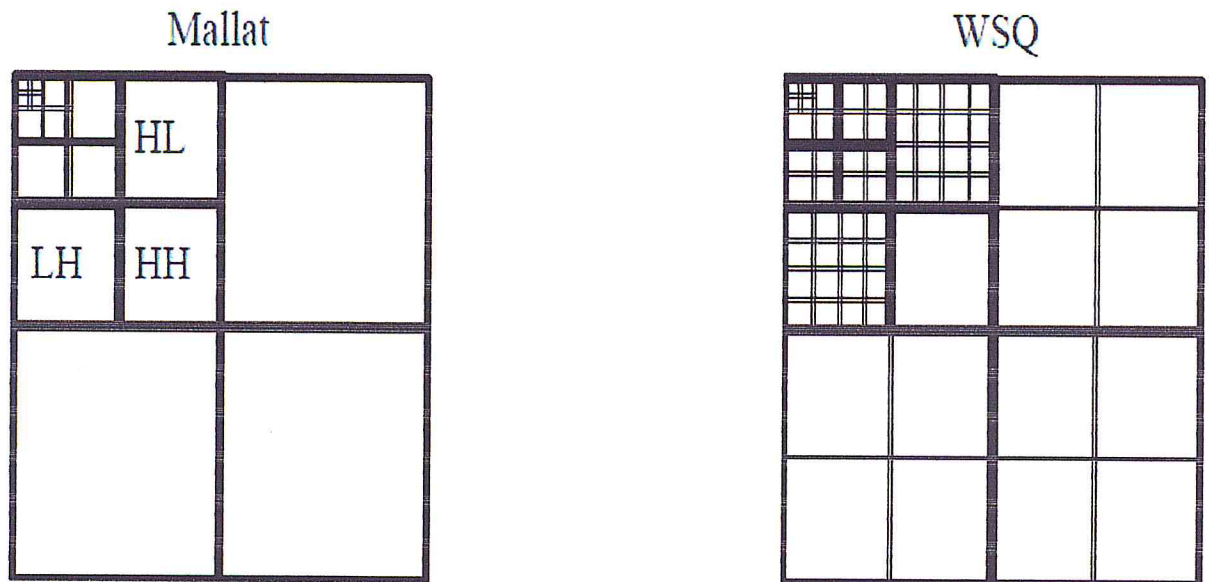


Figure. 3.11 : Décomposition en Ondelettes selon Mallat et WSQ

- Les deux systèmes utilisent la quantification scalaire uniforme ; le pas de quantification en JPEG2000 variant en fonction de la dynamique de la gamme d'activités des sous-bandes. Les étapes de quantification du WSQ sont les mêmes que celles de JPEG sauf pour un intervalle allongé.
- Pour la dernière étape de codage, WSQ emploie le codage entropique de Huffman et le JPEG2000 utilise le codage arithmétique.

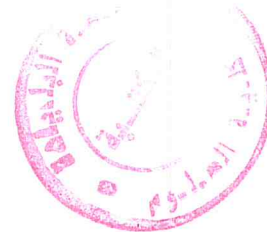
Les études qui ont été faites sur JPEG 2000 et WSQ ont conclu que pour la même taille de fichier, JPEG2000 produit une qualité de reconstruction d'image légèrement inférieure à celle proposée par WSQ [54].

Actuellement, les normes ISO / CEI sur les technologies des formats d'échanges de données biométriques recommandent d'utiliser la méthode WSQ pour les images d'empreintes digitales de 500 dpi avec compression limitée à 15:1, mais pour les images avec plus de 500 dpi, elle recommande d'utiliser JPEG 2000 à 15:1 [42].

Conclusion

La norme de compression par ondelettes WSQ (Wavelet Scalar Quantization) utilise la transformée DWT (Discrete Wavelet Transform). Cette technique traite les images selon le contraste hautes / basses fréquences. Les hautes fréquences concernent principalement les contours des objets et sont ici traitées séparément des basses. On leur applique une compression plus faible mais surtout non destructive. Le reste étant géré de manière plus classique, avec une compression très élevée. Il en résulte une qualité nettement supérieure tout en réduisant davantage la taille des images.

En outre la compression WSQ permet d'éviter l'un des plus douloureux problèmes de JPEG, à savoir la création de blocs de 8x8 pixels. Dès que la compression est importante, ces blocs font apparaître de gros défauts sur les images en regroupant artificiellement les pixels et en donnant cet aspect "mosaïque" désagréable.



IV.1 ETUDE ET PRESENTATION

DU "First-Compression"

1. Introduction:

Dans cette partie nous allons aborder notre réalisation en ce qui concerne les techniques orientée objets utilisées dans l'élaboration de notre application « *First-Compression* ».

Il s'agit dans cette partie d'étudier, de décortiquer, d'analyser et de décomposer la conception de First-Compression, pour en faire ressortir les éléments forts et faibles de logiciel. Merci merci

Les étapes de la démarche adoptée dans notre travail sont les suivantes :

- voir la théorie du signal,
- Etudier la théorie des ondelettes,
- comprendre la théorie de compression d'image,
- Analyse les logiciels de compression disponible,
- Prise de contacts avec les développeurs dans *le domaine de compression*,
- dégager un algorithme de traitement
- Rédiger un software pilote qui implémente l'algorithme.

2. Présentation de *First-Compression* :

First-Compression est application permet de fournir à l'utilisateur un outil qui l'aide à réaliser la compression / décompression des images faciales ou des empreintes digitales d'individus selon le format FBI - Wavelet Squalar Quantization qui se base sur les différents états calculés selon la transformée en ondelettes sur la matrices de pixels d'une image.

3. Description de la structure de *First-Compression* :

La structure de *First-Compression* est organisée selon deux types de classes :

- Les classes du *codeur de First-Compression*,
- Les classes du *décodeur de First-Compression*,

3.1 Les classes du Codeur :

La première étape dans la compression « *First-Compression* » d'une image est la *numérisation*, c'est-à-dire la transformation de l'information que constitue l'image en données numériques. Elle est réalisée au niveau de la classe *Image* par la fonction `void ChargerMatriceImage (int x0, int y0, int x1, int y1)` tel que x_0, x_1, y_0, y_1 représentent les coordonnées de l'image.

Pour une numérisation d'images en noir et blanc :

- dans le cas d'une image faciale nous avons utilisé une transformation de couleur pour obtenir une image au niveau de gris, Elle est réalisée au niveau de *Niveau_du_Gris*.
- mais dans le cas d'une image d'empreintes digitales, elle est déjà au niveau de gris.

Dans la numérisation d'images en noir et blanc (empreintes digitales), il faut tenir compte de deux facteurs : l'intensité : est mesurée selon une échelle de 0 (noir) à 255 (blanc), et la résolution.

On a choisi la résolution des images de 512 X 512 puisque c'est la résolution standard du format WSQ. Mais, dans notre application et avec le logiciel choisi (C Borland), nous avons trouvé des difficultés de l'implémentée car sa résolution est limitée de 250 X 250. Aussi, nous avons utilisé l'allocation dynamique de la mémoire .

Le *codeur First Compression* utilise pour la transformée en ondelettes une banque de filtres numériques correspond à une base d'ondelettes biorthogonales construite par Cohen, Daubechies et Feauveau.

Pour visualiser l'intérêt de ce choix, First-Compression implémente un autre type d'ondelette c'est l'ondelette de Haar.

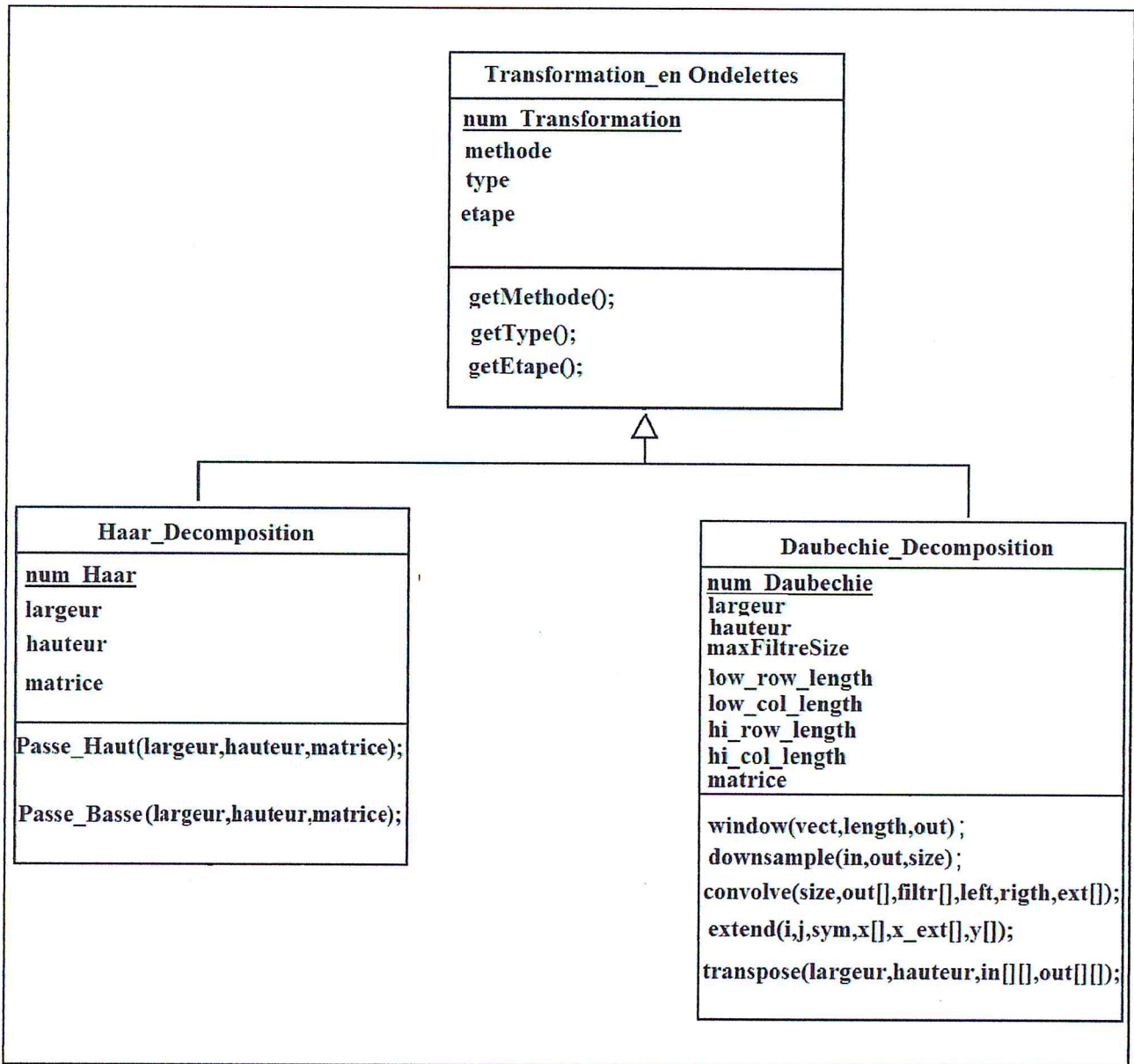


Figure.4-1 : Les deux types de la transformation en ondelettes

C'est ce qui présenter dans le diagramme de séquence suivant :

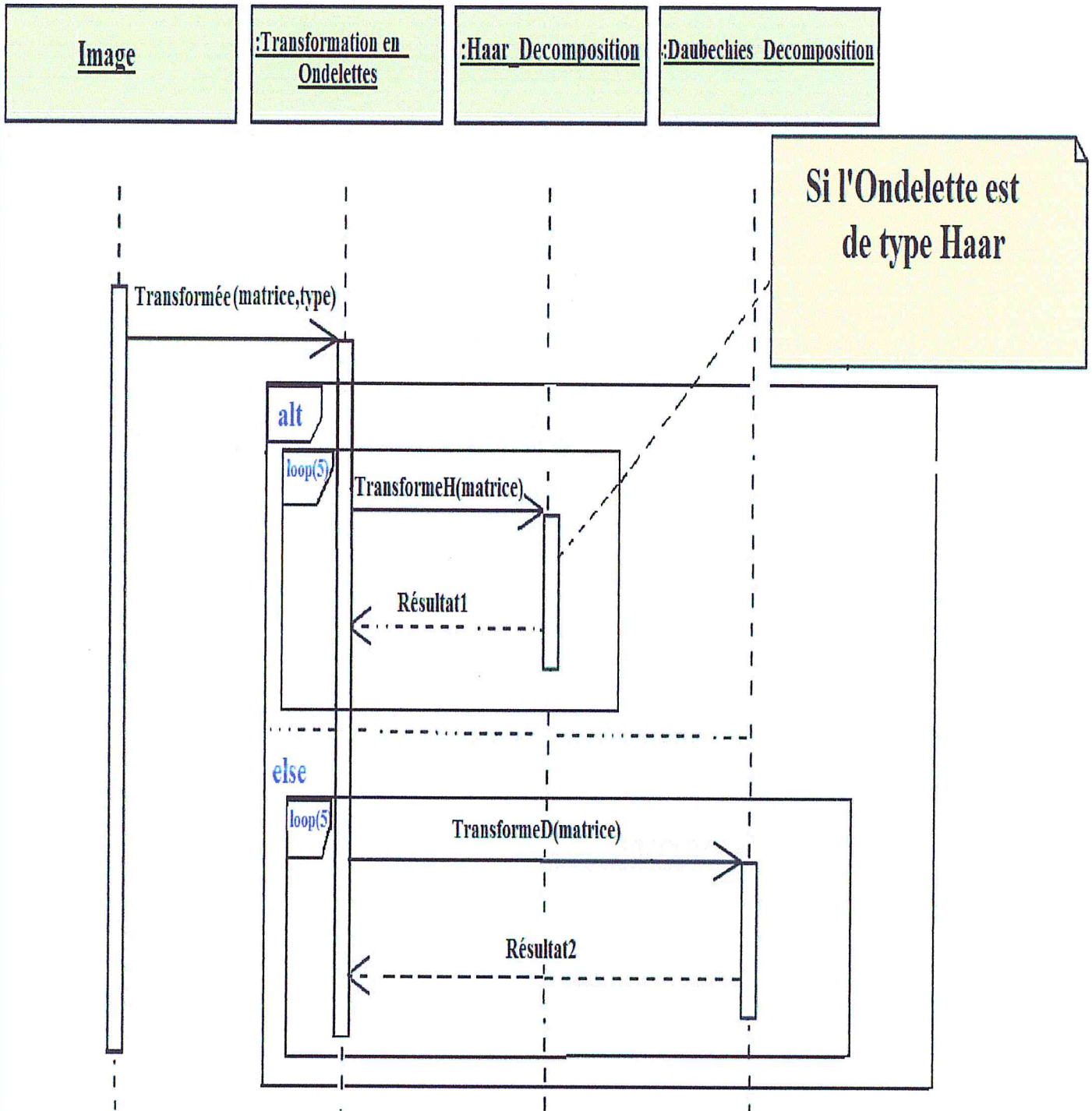


Figure.4-2 : Diagramme de séquence pour les deux types de la transformation en ondelettes

Dans le cas de l'ondelette Daubechies, Nous avons appliqué la phase de normalisation qui permet de meilleurs taux de compression.

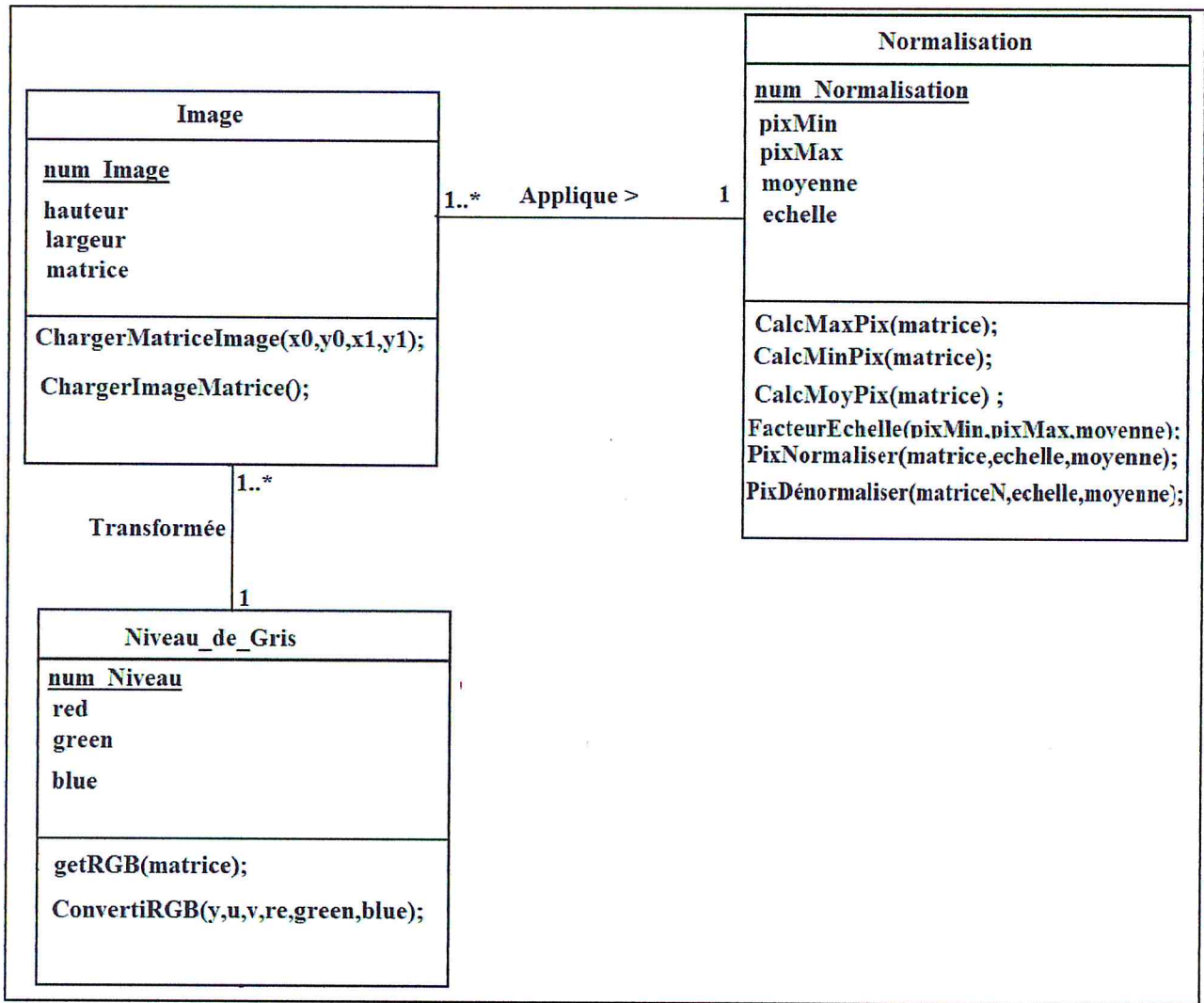


Figure.4-3 : Numérisation d'image

Pour améliorer ces résultats, notre application propose un autre type de filtre. Il s'agit du filtre moyen qui permet de réduire les grands décalages entre les pixels.

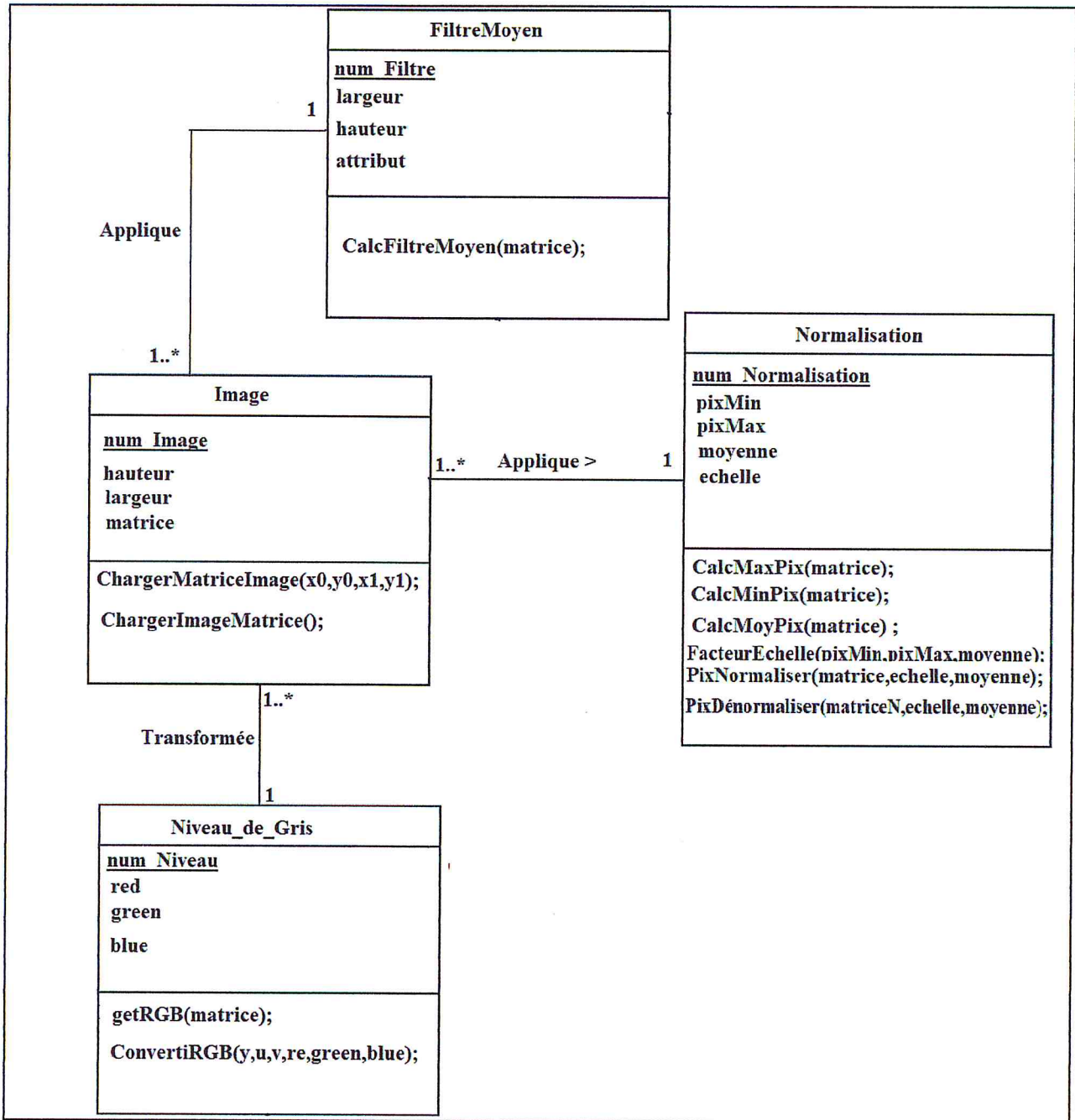


Figure.4-4 : Numérisation d'image avec filtre moyen

Le diagramme suivant décrit la dynamique de l'interaction des objets.

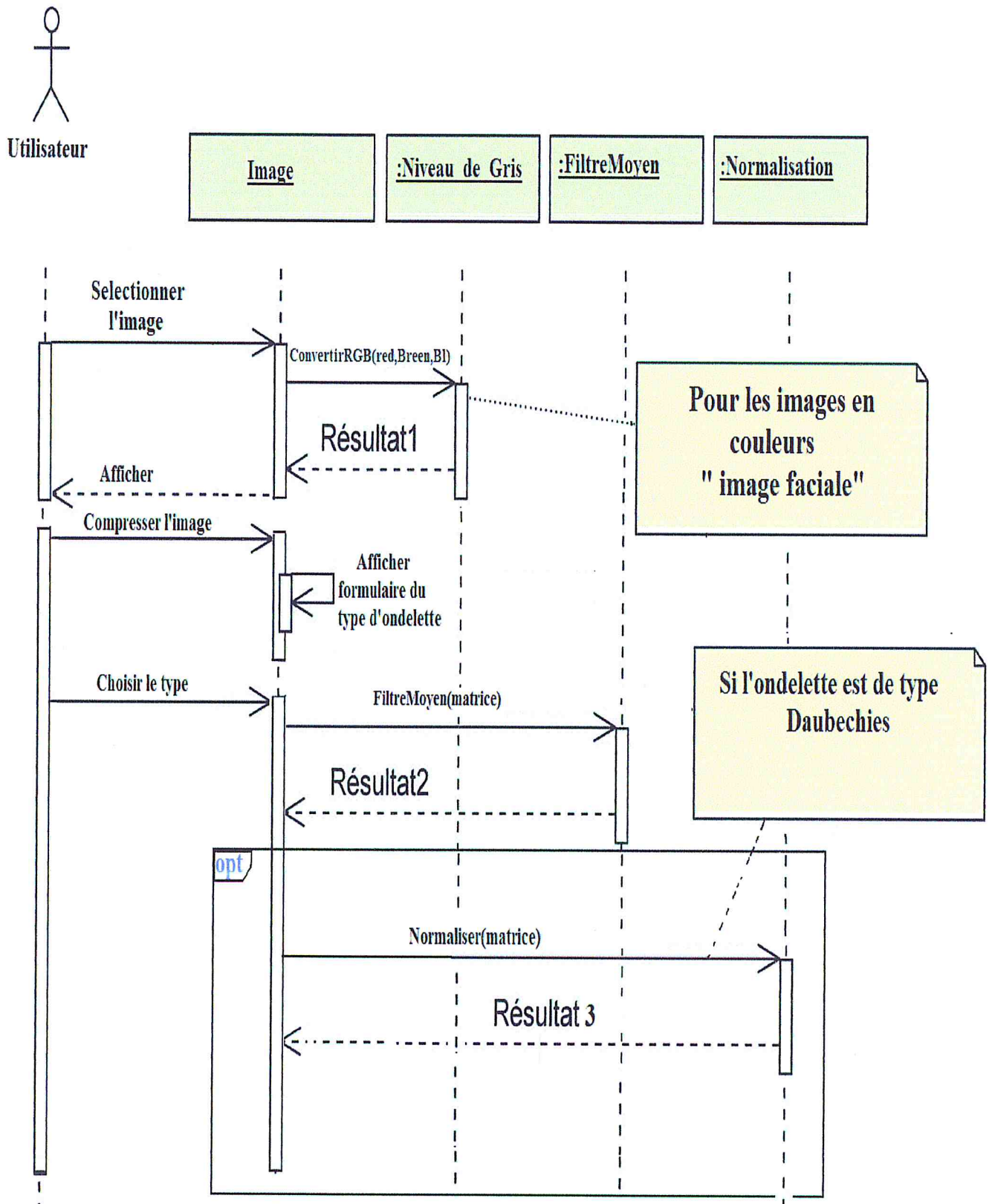


Figure.4-5 : Diagramme de séquence pour numérisation d'image

Dans la classe *Daubechie_decomposition*, nous avons effectué la transformation en ondelettes symétriques. Généralement, on parle de la transformation en ondelettes discrètes mais pourquoi symétriques ?

Parce que si la fonction d'échelle et l'ondelette sont symétriques ou antisymétriques, alors les filtres sont à phase linéaire. Si on n'a pas cette propriété, cela entraîne une distorsion de phase lors de la reconstruction.

First-Compression utilise les 2 classes de filtres de réponse impulsionnelle finie à phase linéaire (RIF) définie dans la norme WSQ pour la décomposition dans la classe *Daubechie_decomposition* :

- La classe *Filtre_Impair9/7* : est le filtre symétrie impair (odd-length)
- La classe *Filtre_Pair* : est le filtre pair (even-length). De sorte, le filtre passe-bas de cette classe est symétrique et le filtre passe haut est antisymétrique.

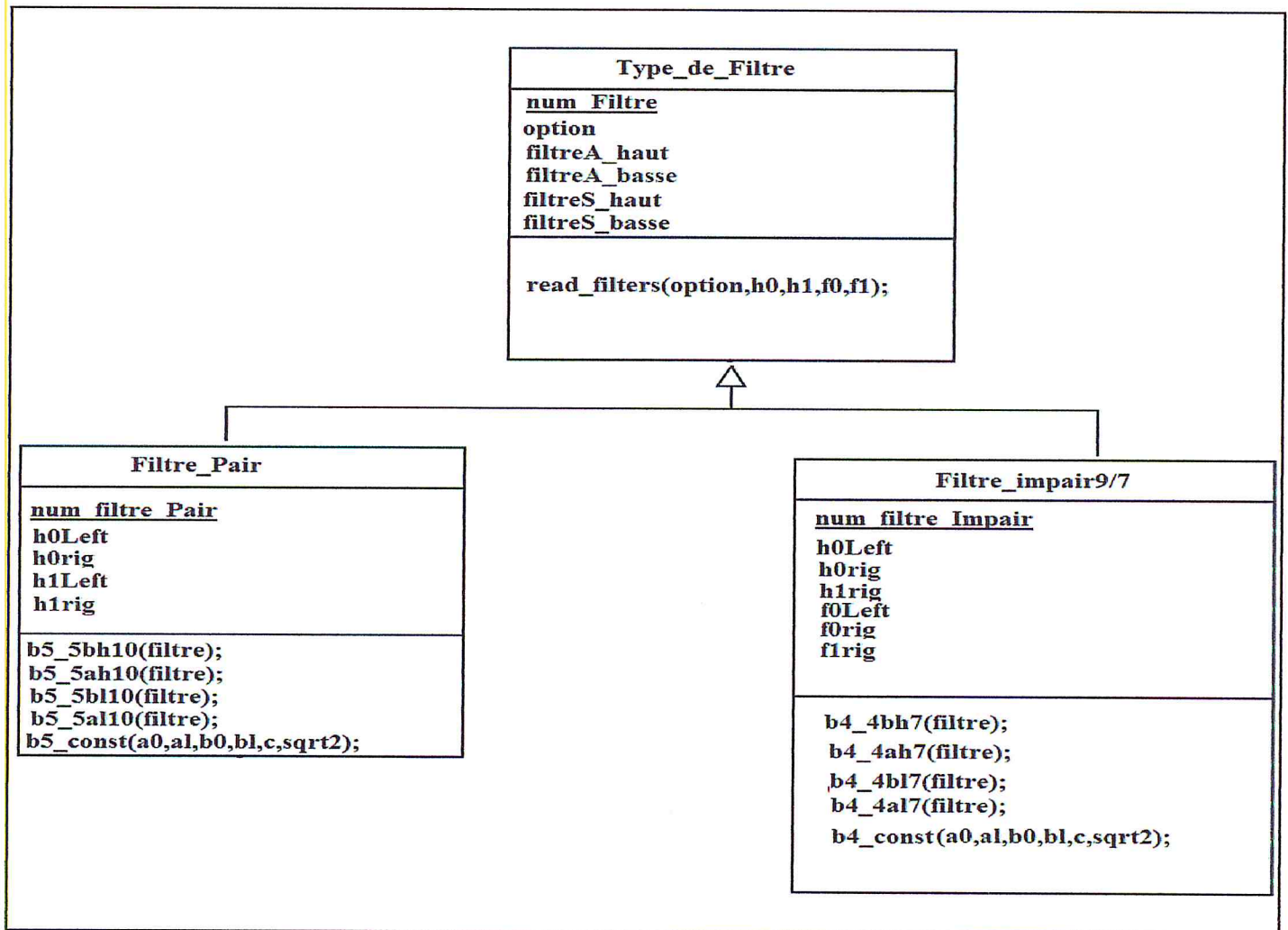


Figure.4-6 : Les deux types de filtres

Ces filtres ont été sélectionnés par des tests des erreurs des performances du quantificateur, en comparaison avec d'autres banques de filtres à reconstruction parfaite ; cette banque de filtres particulière a produit une quantification d'images supérieures à celles générées par les autres filtres testés. Le fait que le support de ces filtres corresponde de très près à la fréquence des crêtes des empreintes digitales est probablement un autre facteur de leurs excellentes performances sur des images d'empreintes digitales.

La transformation est réalisée en plusieurs étapes qui consistent à :

- Créer une extension symétrique pour chaque vecteur de la matrice d'image,
- Appliquer une convolution entre l'extension et le filtre passe « passe-bas ». C'est la partie la plus importante avec des composants à grande échelle. Généralement on les appelle « *approximations* »,
- L'exécution de filtrage multiplie la quantité de données du départ par deux. C'est pour cette raison que nous avons utilisé la notion de *sous-échantillonnage* sur le résultat qui consiste en le rejet de toutes données secondaires,
- Appliquer une convolution entre l'extension et le filtre « passe-haut ». C'est la partie qui permet d'ajouter un goût ou une nuance à l'extension. Généralement on l'appelle « détail »,
- Faire le *sous-échantillonnage* sur cette extension,
- Placer les coefficients passe-bas dans la première partie du vecteur et les coefficients passe-haut dans la seconde partie,
- Transposer la matrice pour refaire ces étapes sur les colonnes de l'image.

La procédure de la transformation en ondelettes est répétée plusieurs fois jusqu'à produire 64 sous-bands. Son intérêt est qu'elle permet d'améliorer la compression car elle se rapproche de l'orthonormalisation.

La phase de quantification de l'ondelette Daubechies est réalisée par la classe *Quantification* tel qu'on applique une quantification uniforme sur chaque sous-image.

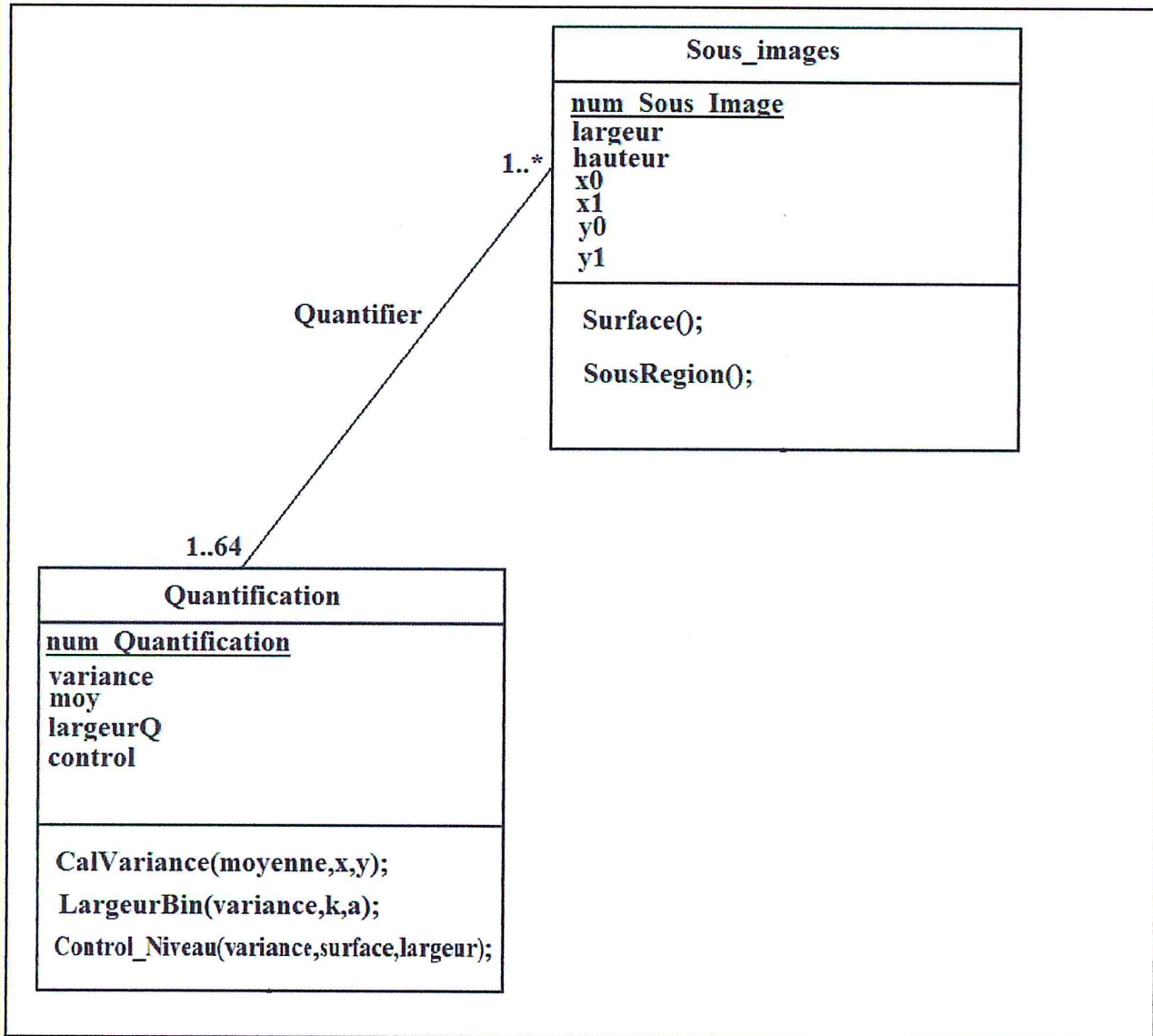


Figure.4-7 : Quantification de l'ondelette Daubechies

Le diagramme de séquence suivant montrant la transformation selon l'ondelette de Daubechies :

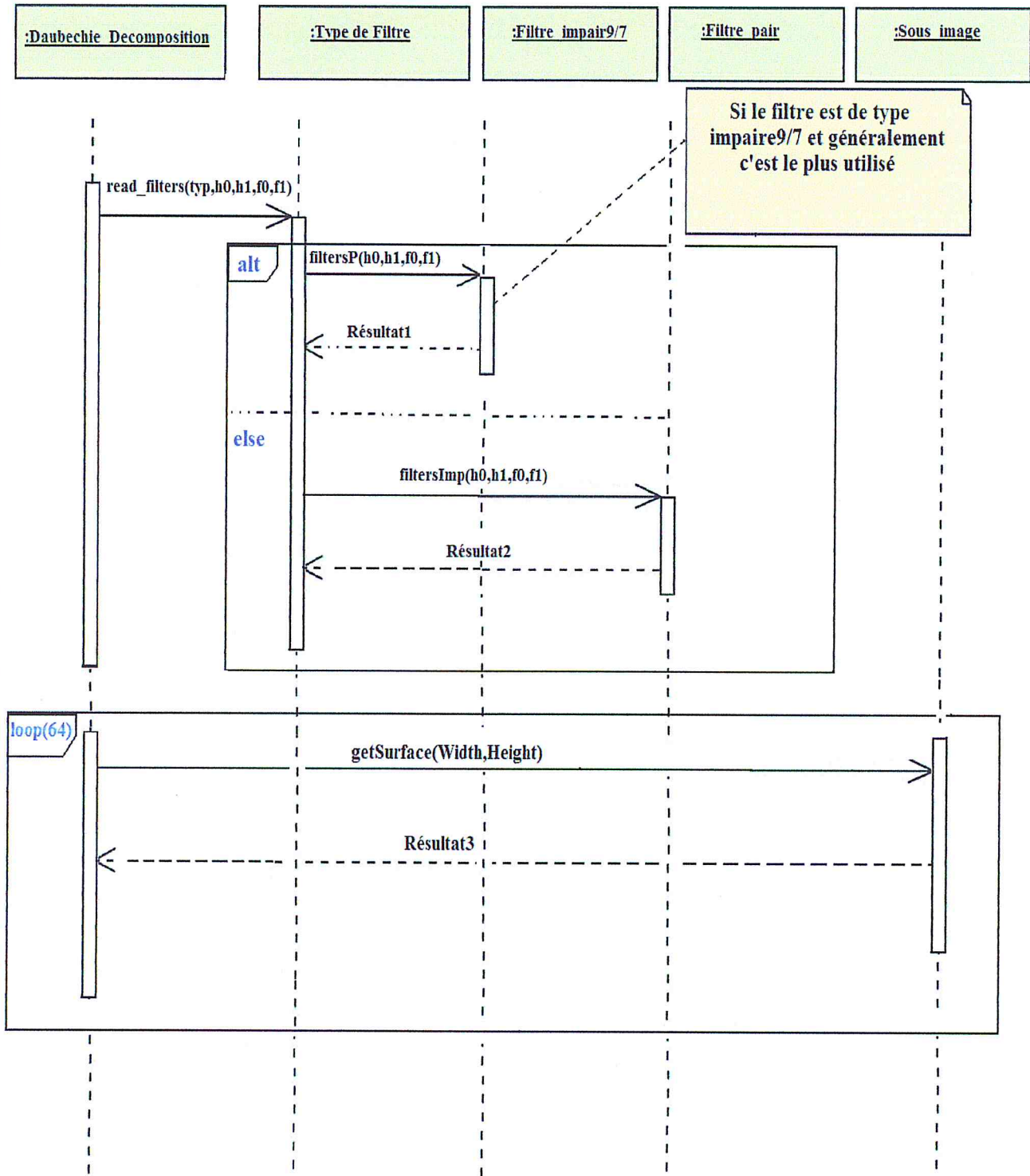


Figure.4-8 : Diagramme de séquence pour la transformation par l'ondelette de Daubechies

La phase de codage est réalisée en deux étapes :

- La première permet au codeur de construire des blocs de codage entropique et d'appliquer sur chaque bloc le codage run-length en utilisant la table de Huffman suivante :

position	value
1	zero run length 1
2	zero run length 2
3	zero run length 3
⋮	
100	zero run length 100
101	esc for pos 8 bit coeff
102	esc for neg 8 bit coeff
103	esc for pos 16 bit coeff
104	esc for neg 16 bit coeff
105	esc for zero run - 8 bits
106	esc for zero run - 16 bits
107	coeff value -73
108	coeff value -72
109	coeff value -71
⋮	
180	- use position 1 only -
⋮	
253	coeff value 73
254	coeff value 74

Figure.4-9 : Tableau d'Huffman

Ce tableau permet d'éviter le doublage du Zéro et de remplacer les grands nombres par des petits nombres, c'est-à-dire de réduire l'espace de stockage.

➤ La deuxième étape permet d'appliquer le codage Huffman.

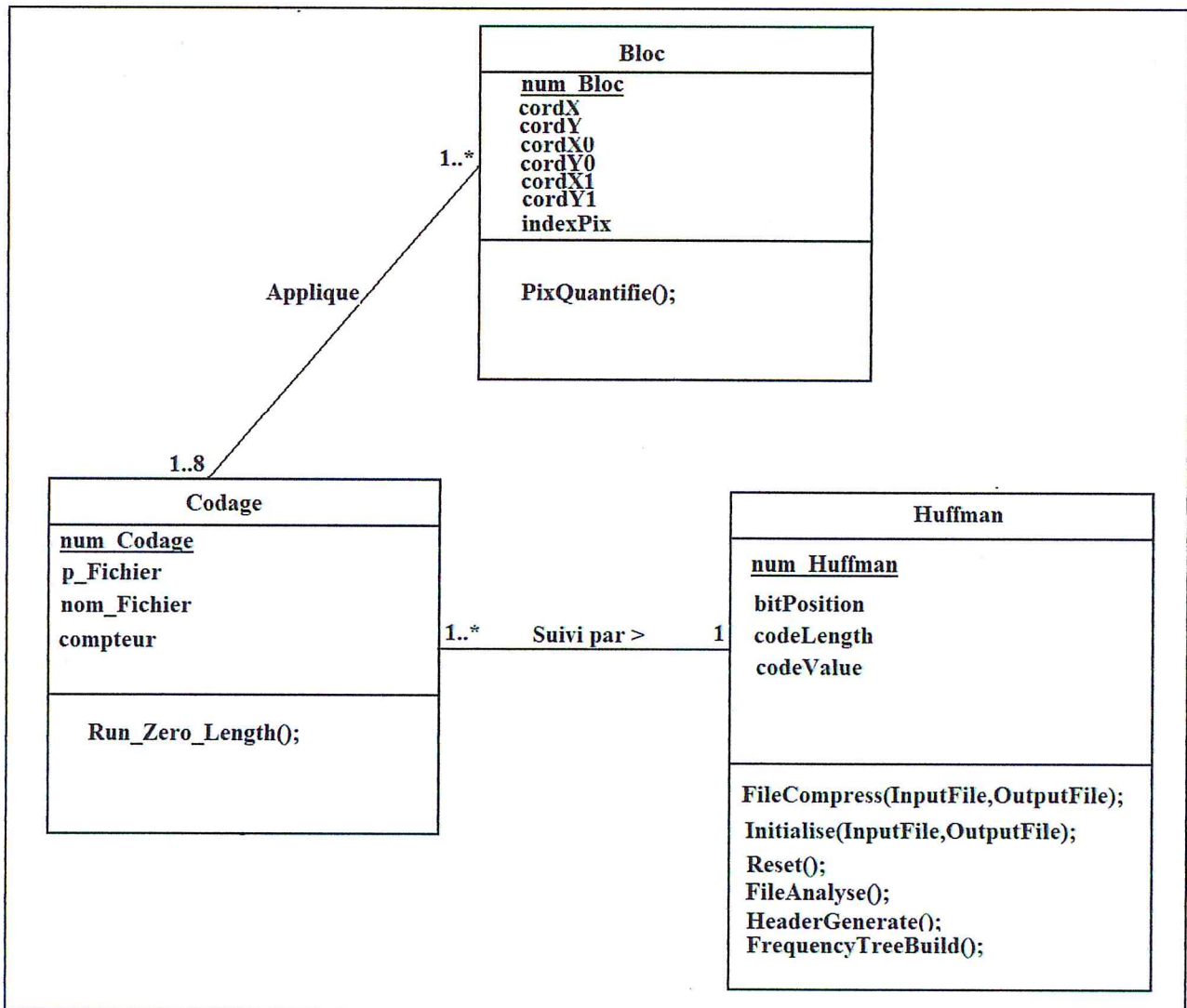


Figure.4-10 : Codage de l'ondelette Daubechies

Pour la classe *Haar_Decomposition* présente la transformée en ondelette de Haar, elle applique l'arbre de décomposition dyadique basé sur le schéma de Mallat, la quantification est simple elle est réalisée par la classe *Haar_Quantification*.

Le codage de cette ondelette utilise les deux types de codage.

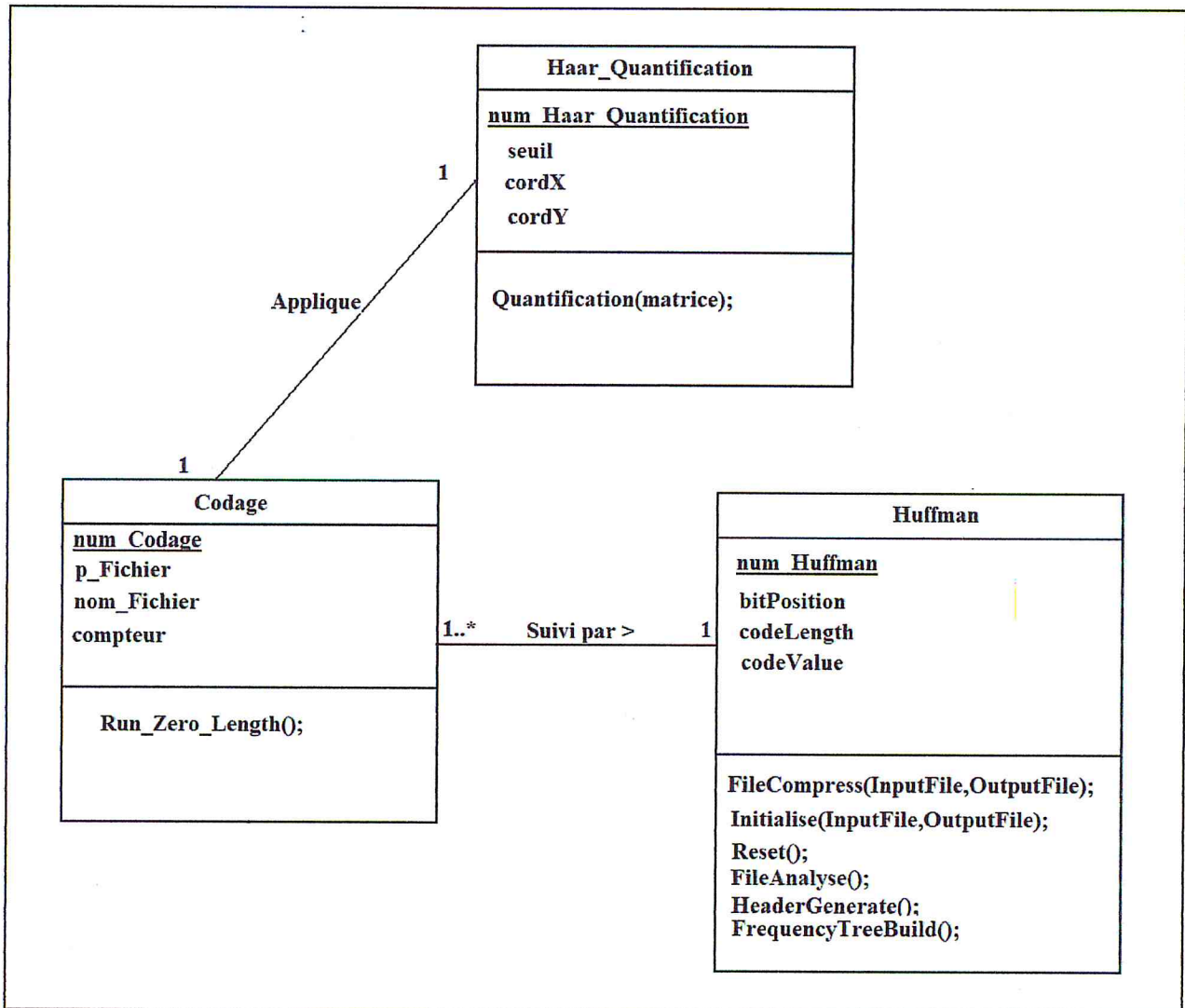


Figure.4-11 : Quantification et Codage de l'ondelette Haar

Le diagramme de classe suivant, donne une vue statique sur les classes qui entrent dans l'application First Compression.

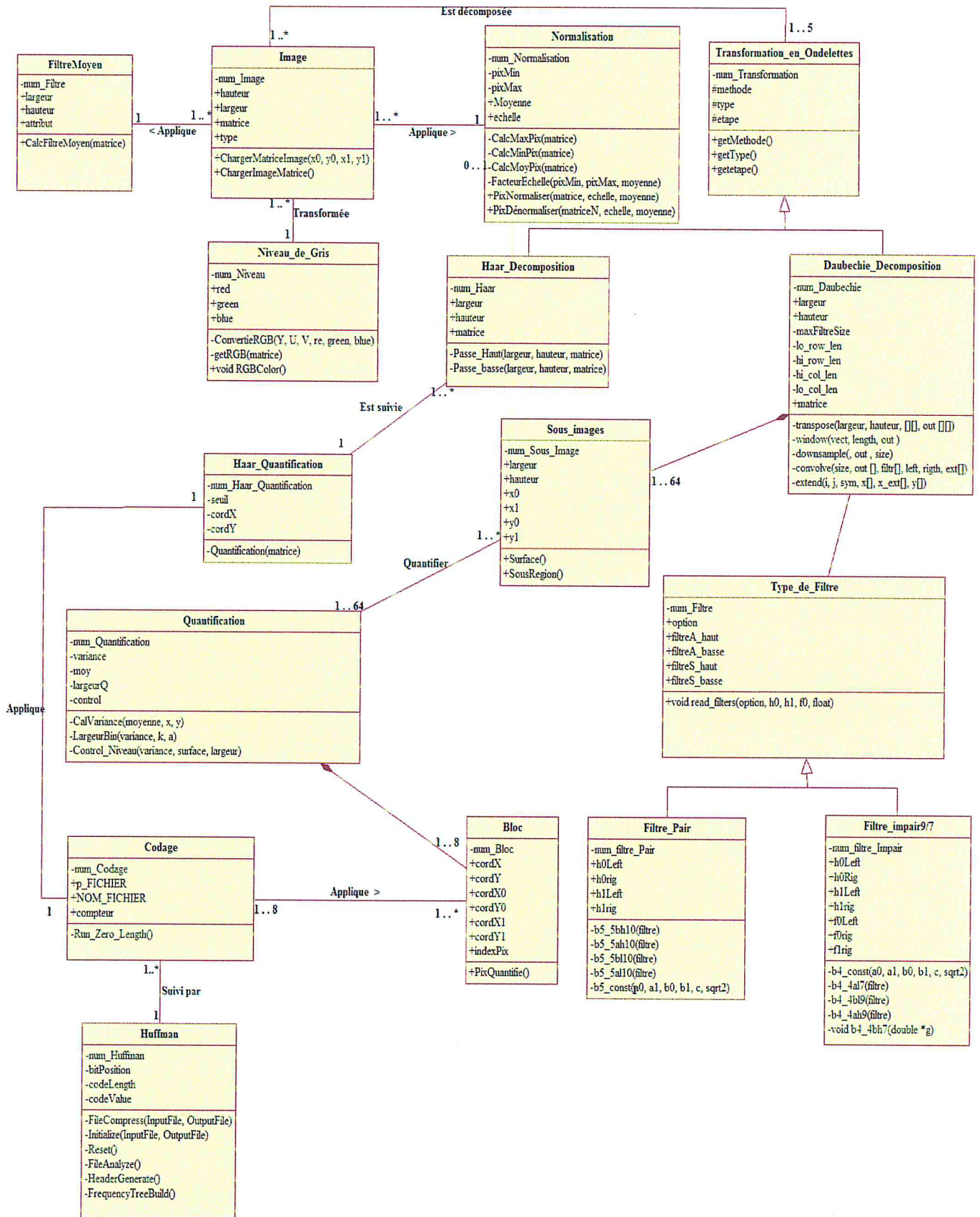


Figure.4-12: diagramme de classe pour le Codeur de First Compression.

On va expliquer dans le diagramme de séquence suivant l'enchaînement des phases de codage, pour montrer du point de vue dynamique comment s'effectue la compression selon le format WSQ.

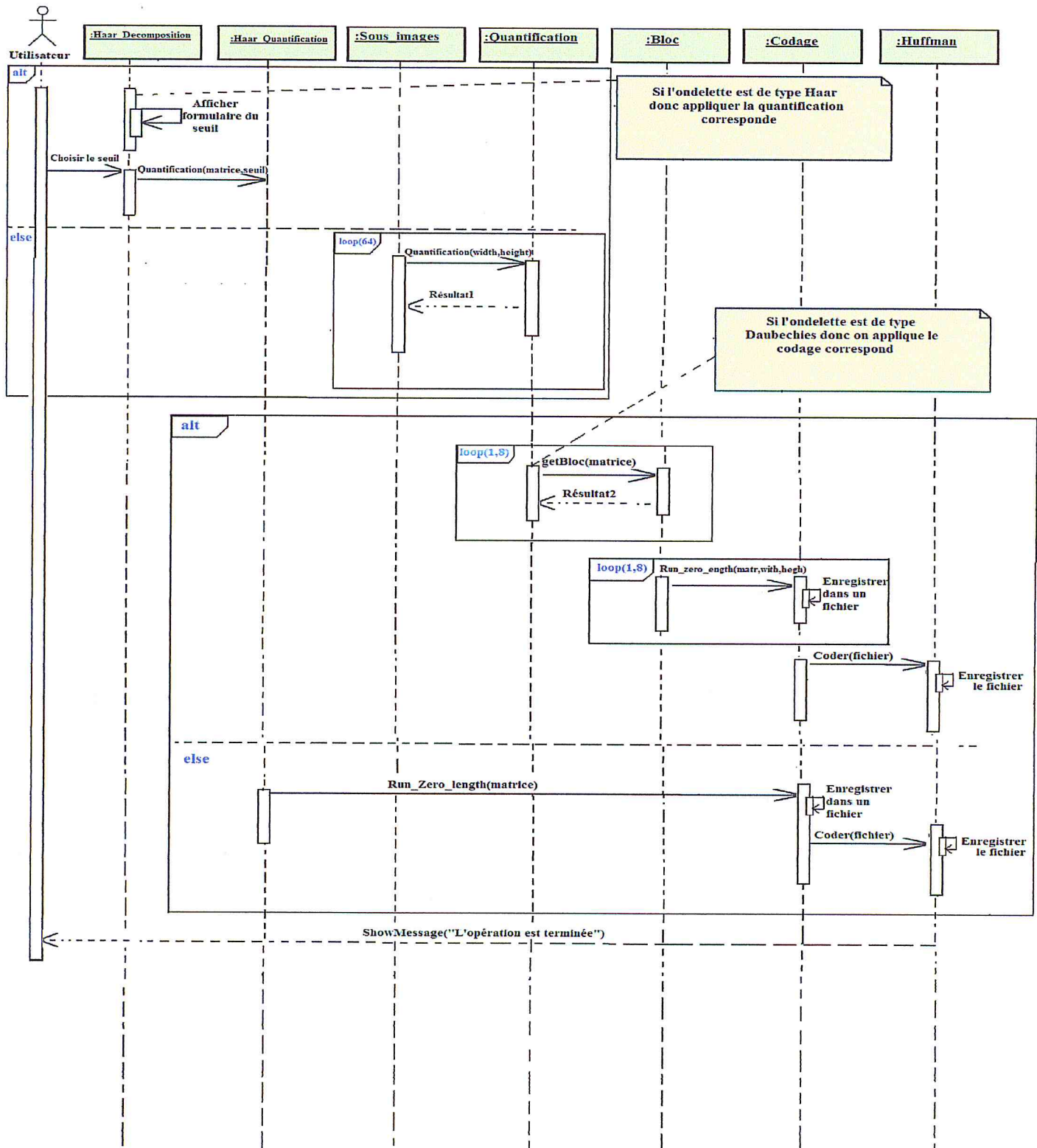


Figure.4-13: diagramme de séquence pour le Codeur de First Compression.

3.2 Les classes du Décodeur :

En commence par décoder le codage huffman (*la classe Decod_Huffman*) puis par reconstruire les coefficients, on applique une transformation inverse du tableau : par exemple, au lieu d'écrire 100, on écrit cent fois 0 (*la classe Codage_Inverse*).

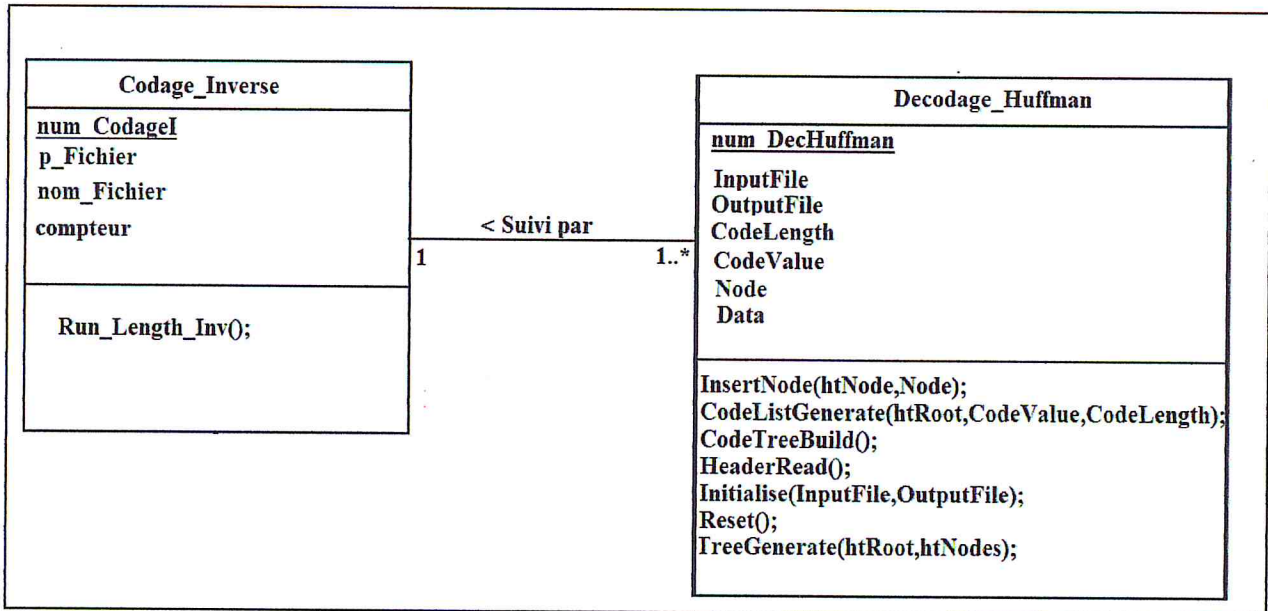


Figure.4-14 : Codage Inverse

La quantification inverse en *First Compression* est réalisée selon le type d'ondelette utilisée :

- Dans le cas d'ondelette Daubechies on utilise *la classe Quantification_Inv_Daubechies*.
- Dans le cas d'ondelette Haar on utilise *la classe Quantification_Inv_Haar*.

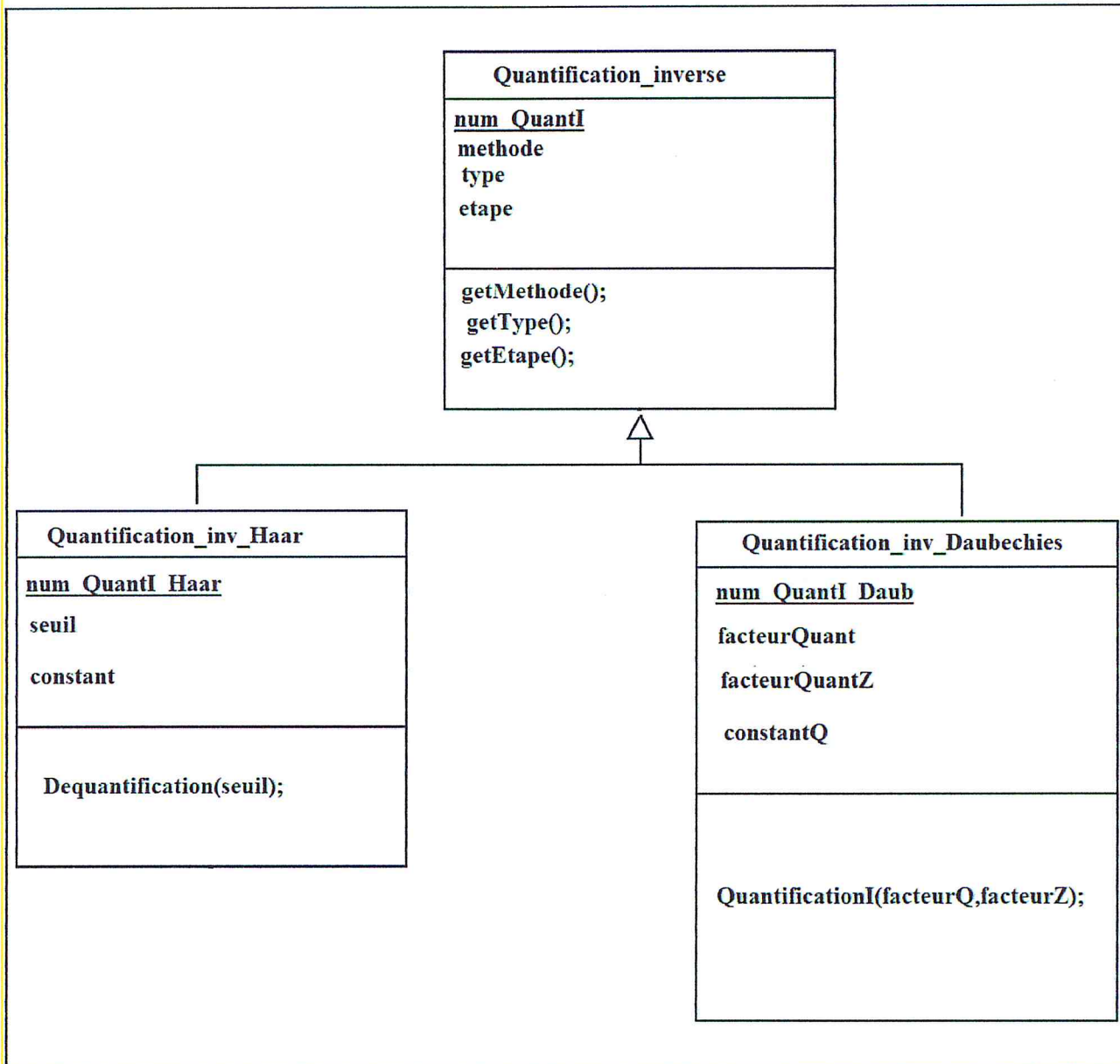


Figure.4-15 : Quantification Inverse

- Placer les coefficients passe-bas de synthèse dans la première partie du vecteur et les coefficients passe-haut dans la seconde partie. Cette fois, le processus s'effectue de façon inverse au processus de codage et commence par le sous-band 64 pour arriver au sous-image 1.

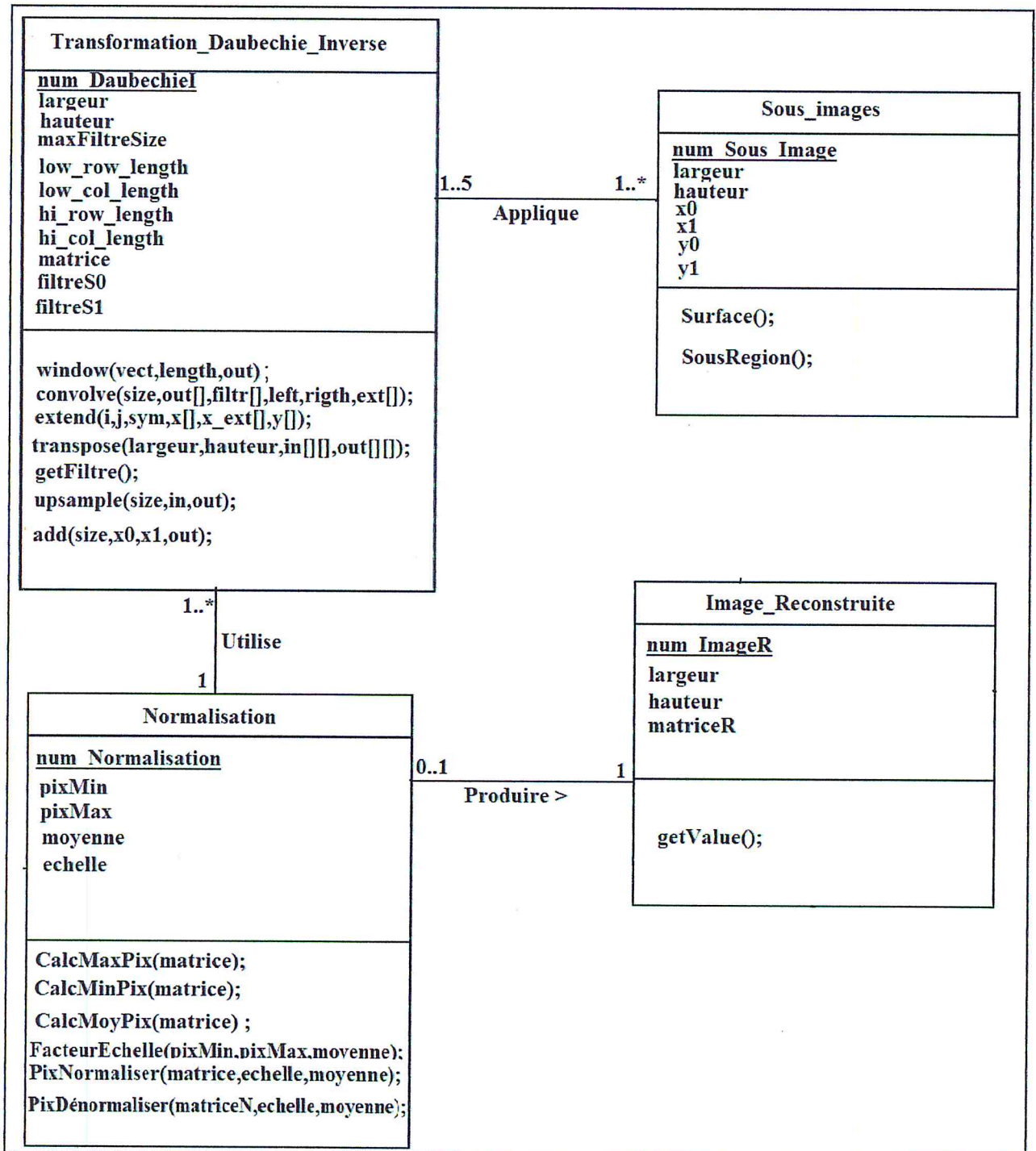


Figure.4-17 : Transformation Inverse de l'ondelette Daubechies

La dénormalisation devient indispensable pour reconstruire l'image.

Le diagramme suivant décrit la dynamique de reconstruction d'image dans le cas d'ondelette de Daubechies.

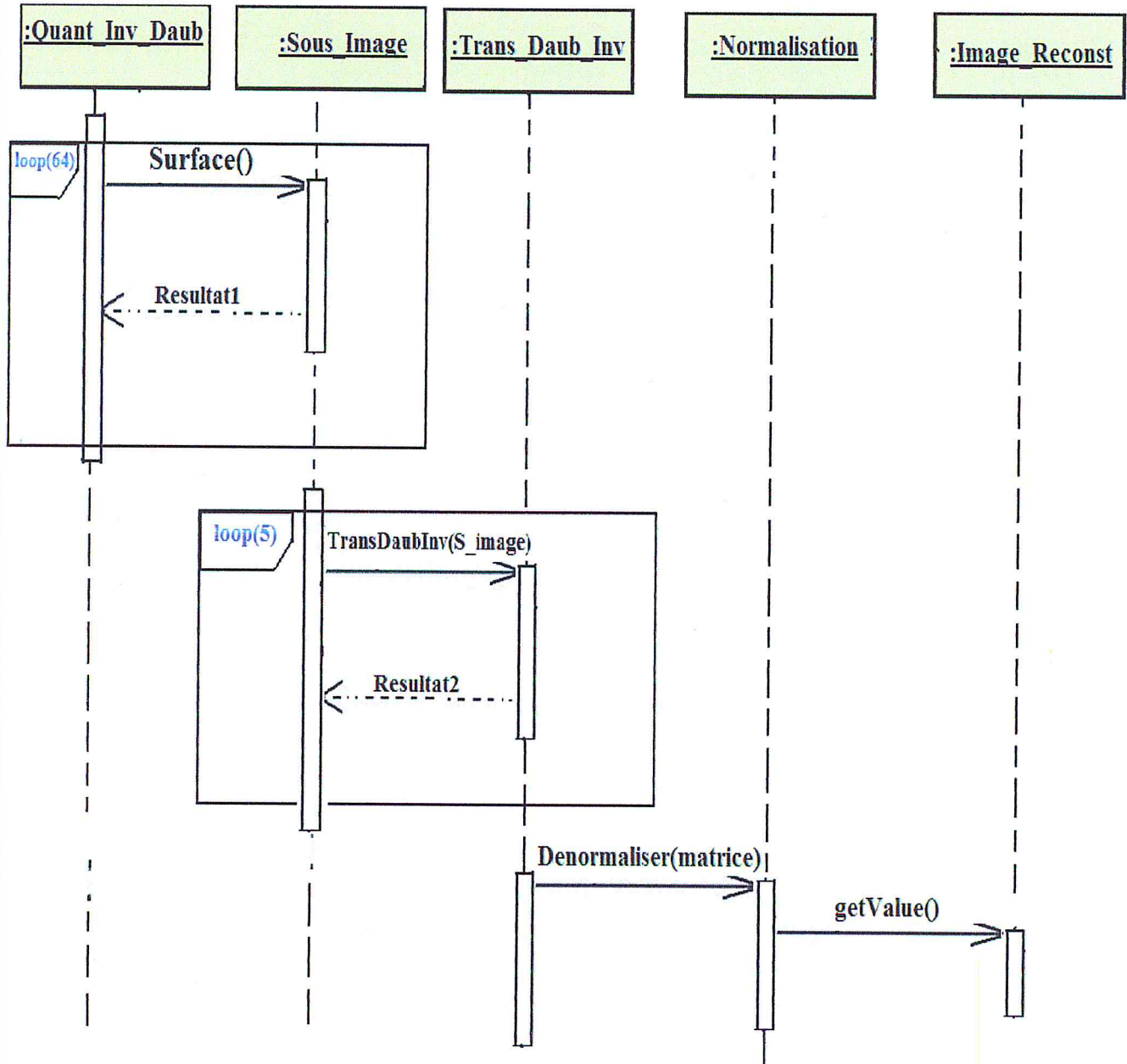


Figure.4-18 : Diagramme de séquence pour l'image reconstruite par l'ondelette de Daubechies

La transformation en ondelettes inverse de Haar est réalisée par la classe *Transformation_Inverse_Haar* a fin de reconstruire l'image.

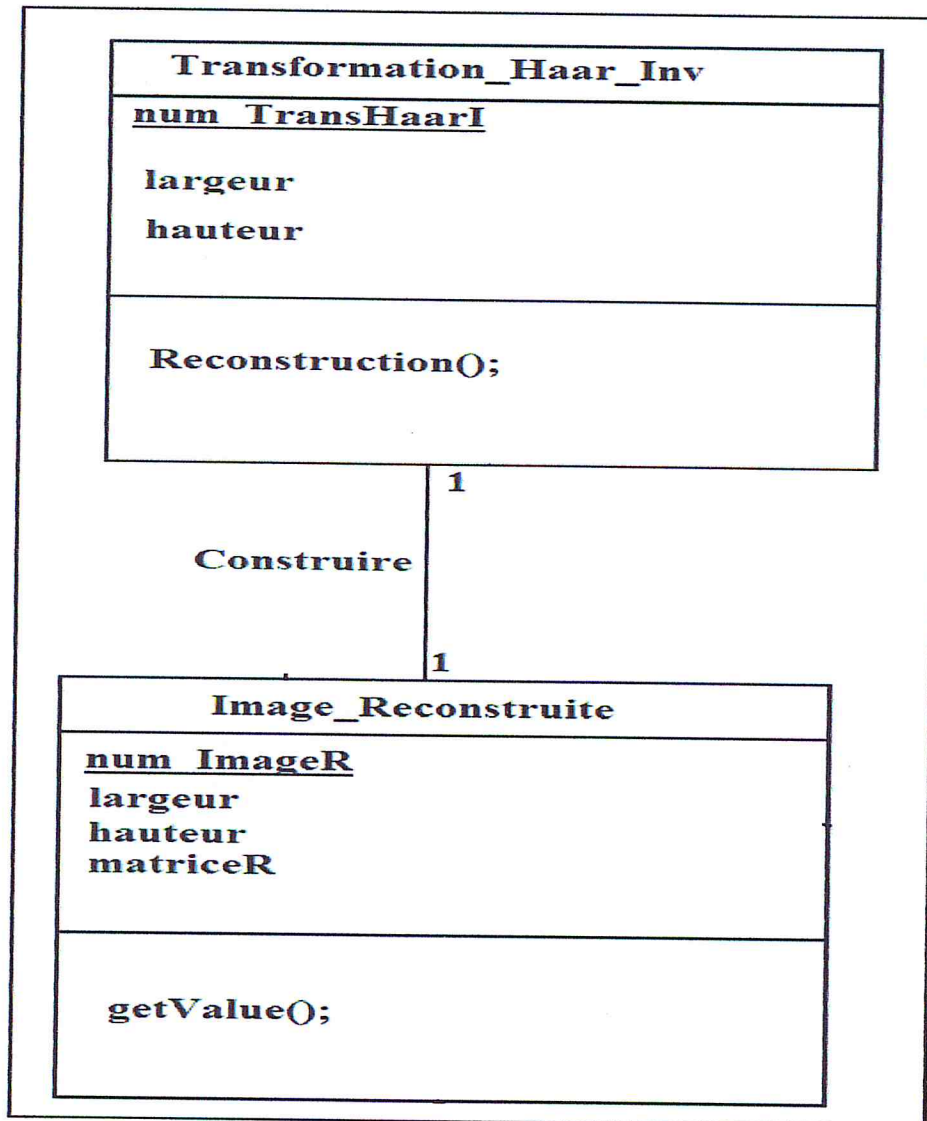


Figure.4-19 : Transformation Inverse de l'ondelette Haar

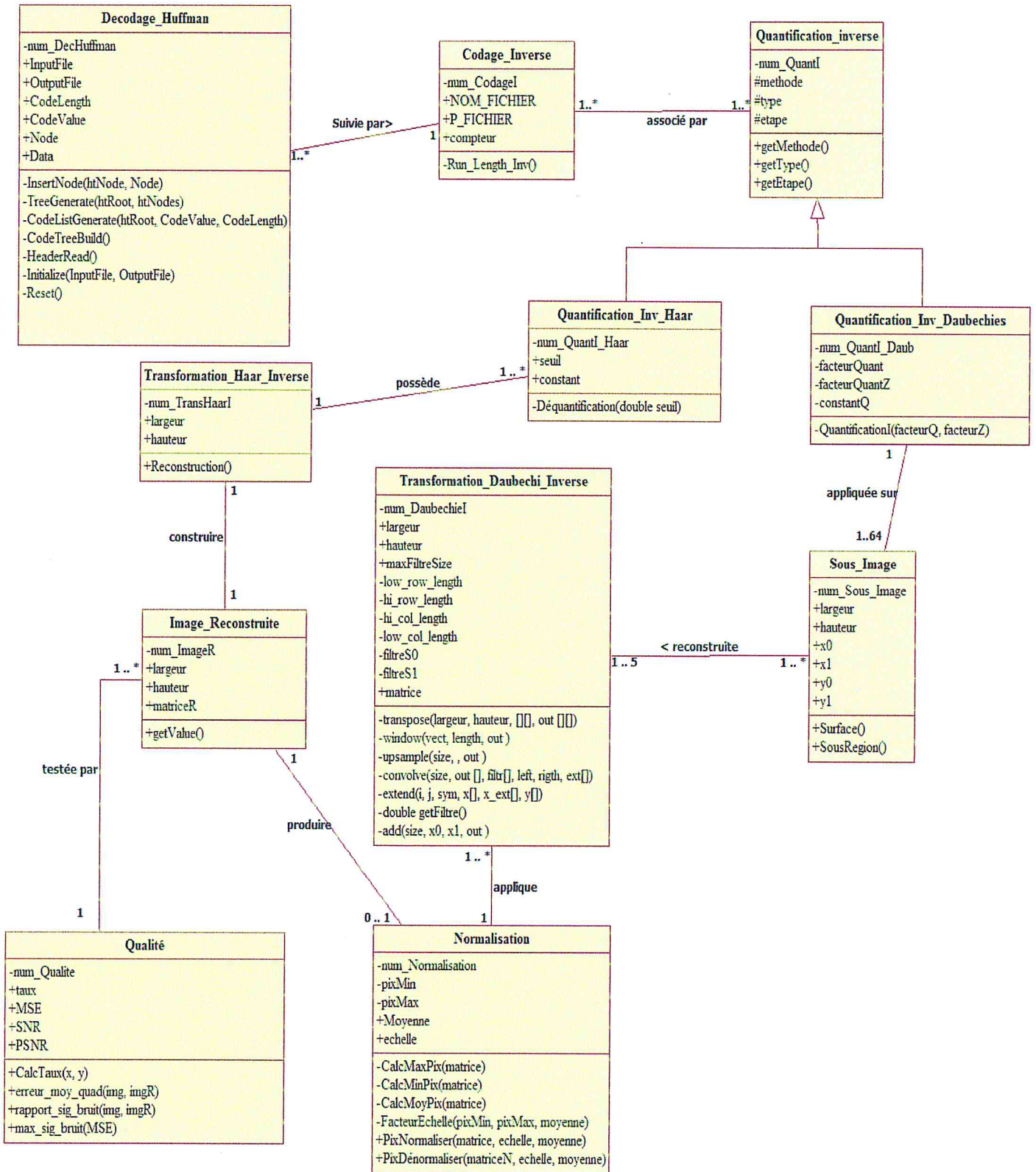


Figure.4-20 : diagramme de classe pour le Décodeur de First Compression.

VI.2 TESTS EN FIRST COMPRESSION

1. INTRODUCTION

Les tests dans la maintenance de logiciel sont effectués durant le développement, il va permettre la correction de défauts existants, et ceux découlant de l'ajout de fonctionnalités.

Lors du développement de First Compression nous sommes passés par plusieurs étapes, et dans chacune d'elles, nous avons dépensé un temps plus important que celui du développement à faire des tests.

Pour vérifier la qualité d'image, on utilise plusieurs facteurs :

a) Le taux de compression :

On mesure l'efficacité de la compression selon la formule suivante :

$$T_c = \frac{\text{Nbre de bits avant compression}}{\text{Nbre de bits après compression}}$$

T_c est toujours plus grand que 1 et nous souhaitons qu'il soit le plus grand possible.

b) Les mesures de fidélité :

Il s'agit de définir des quantités permettant d'évaluer numériquement la qualité de l'image reconstruite. Si l'on note x l'image originale de taille $M \times N$, et \hat{x} l'image de même taille obtenue après reconstruction, on peut définir :

➤ L'erreur moyenne quadratique (MSE : Mean Square Error) :

MSE calculée en considérant les écarts existant entre les deux types d'image.

$$MSE = \frac{1}{MN} \sum_{k=1}^M \sum_{l=1}^N [x(k, l) - \hat{x}(k, l)]^2$$

➤ Le rapport signal sur bruit (SNR : Signal to Noise Ratio) : Le critère de distorsion SNR permet de pallier à l'absence de l'œil du spécialiste, mesuré en décibel (dB),

$$SNR_{dB} = 10 \log_{10} \left(\sum_{m,n} I_{m,n}^2 / \sum_{m,n} (I_{m,n} - I'_{m,n})^2 \right)$$

- Le maximum du rapport signal sur bruit (PSNR : Peak Signal to Noise Ratio) :
L'une consiste à évaluer visuellement la qualité des images et l'autre repose sur le calcul du rapport signal sur bruit (PSNR) obtenu en valeurs de crête entre l'image originale et celle qui a été reconstituée, mesuré en décibel (dB) :

$$PSNR = 10 \log_{10} \text{ dynamique de l'image}^2 / MSE^2 \quad (dB)$$

Etant donné qu'une image est généralement codée sur 8 bits et est représentée par 256 niveaux de gris qui varient entre 0 et 255, l'étendue ou la dynamique de l'image est alors de 255. D'où :

$$PSNR = 10 \log_{10} (255 / MSE)^2 \quad (dB)$$

La classe qui permet de réaliser ces tests est la classe *Qualité*.

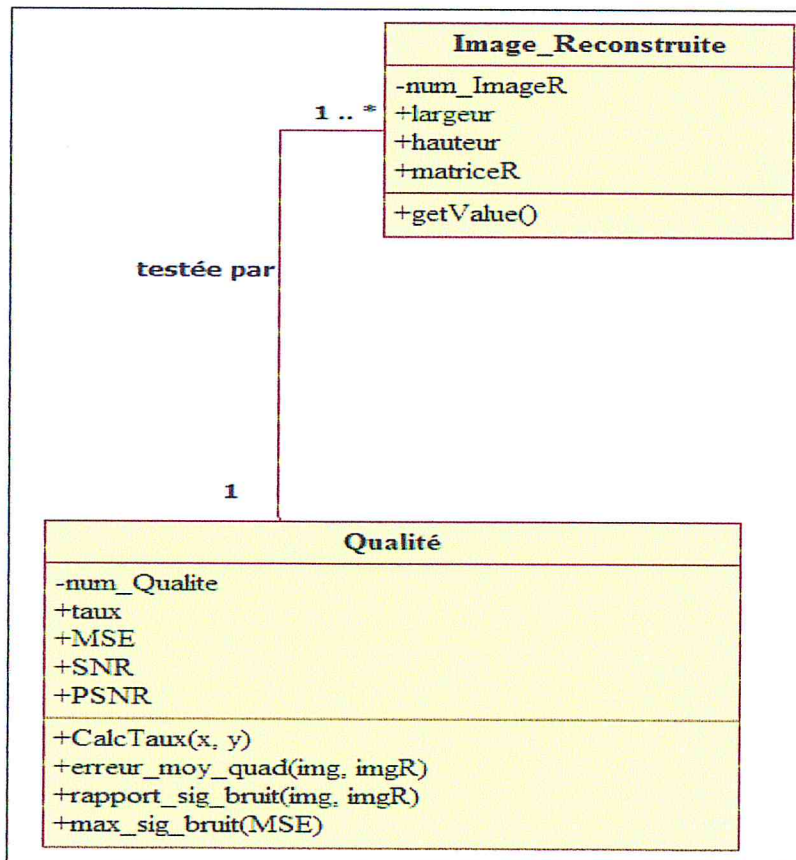


Figure.4-21 : Calculer la qualité d'image reconstruite

Les tests permanents nous ont permis de répondre aux besoins de manière dynamique et de consolider certains choix entrepris.

IV.3 PRÉSENTATION DE L'APPLICATION

1. Quelques interfaces réalisées :

Notre application permet de fournir à l'utilisateur un outil qui l'aide à réaliser la compression / décompression des images faciales ou des empreintes digitales d'individus selon le format FBI - Wavelet Squalar Quantization qui se base sur les différents états calculés selon la transformée en ondelettes sur la matrices de pixels d'une image.



Figure. 4.22 : L'interface d'accueil

Pour commencer l'étape de compression :

- La première étape dans la compression consiste à sélectionner l'image :



Figure 4.23 : L'interface pour sélectionner l'image

➤ La deuxième étape permet d'appliquer la Transformée en Ondelettes Symétrique



Figure 4.24 : L'interface d'image sélectionnée

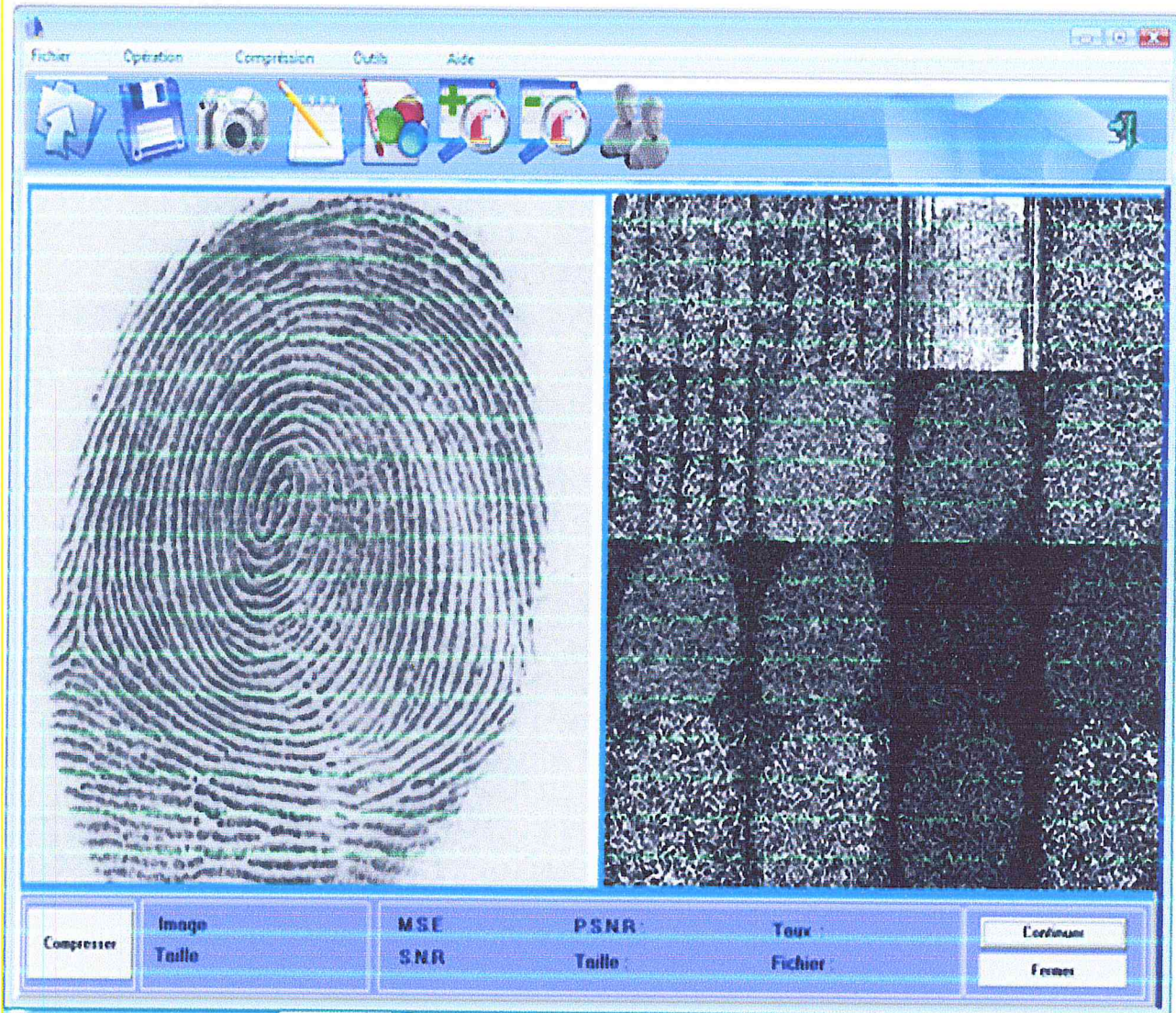


Figure 4.25 : L'interface de codage FBI-WSQ dans le niveau 5

La transformation en ondelettes symétriques est appliquée à une numérisation d'image bidimensionnelle en transformant d'abord les lignes puis les colonnes de l'image, entraînant une décomposition en quatre sous-images. Les quatre sous-images sont à leur tour décomposées en cascade de quatre sous-images chacune pour produire une banque de plus de 16 sous-images. La cascade est répétée plusieurs fois sur certaines des sous-images passe-bas jusqu'à atteindre une décomposition en 64 sous-images, ainsi qu'on peut le voir en figure 4.25. Cette décomposition a été conçue à partir d'une analyse de la puissance et de la densité spectrale (PSD) des empreintes digitales, de la capacité de conditionnement des informations des différentes décompositions, et des études empiriques des effets de la quantification des sous-images sur la qualité de l'image reconstituée.

- Appliquer la quantification scalaire uniforme selon le format FBI-WSQ :



Figure 4.26 : L'interface de codage FBI-WSQ pour la quantification

La compression avec perte de notre application est réalisée par quantification scalaire uniforme sur les sous-images. Les coefficients d'ondelettes sont considérés comme des entrées analogues par le processus d'encodage. Le terme d'amplitude de quantification, ou tout simplement de quantification, fait référence à la procédure de transformation du coefficient d'ondelettes de type réel en l'une des nombreuses valeurs quantifiée \hat{a} . Celle-ci se fait en deux étapes :

- Dans l'encodeur, le coefficient d'ondelette a sera transformé en indice de quantification de type entier p . Une quantification de l'encodeur est décrite mathématiquement par une fonction $E: \mathbb{R} \rightarrow \mathbb{S}$, où \mathbb{S} est un ensemble discret des indices de quantification. L'indice de quantification sera encodé par l'encodeur entropique puis transmis sous un format compressé. Comme E n'est pas inversible, le résultat est, par nature, une compression avec perte.
- Dans le décodeur, il y a une quantification inverse sur les indices de quantification p pour produire un ensemble discret de valeurs reconstruites à virgule flottante, \hat{a} et qui sont

désignées par les termes de coefficients d'ondelettes quantifiés. La quantification par décodeur est décrite mathématiquement par la fonction $D: S \rightarrow R$. Notons que le décodeur, D , n'est pas l'inverse (au sens mathématique) de l'encodeur, E . La fonction composite $F = D \circ E$, qui comprend tout le processus de passage des coefficients d'entrée jusqu'aux coefficients de sortie, est connue comme caractéristique de la quantification.

Au sein d'une même sous-image, les intervalles de quantification sont d'égale largeur, à l'exception de la boîte contenant l'origine (la zone zéro), qui est un peu plus large du fait des estimations du seuil de bruit. La largeur des Bin varie d'une sous-image, k , à une autre. On a choisi deux types de largeurs des zones, q qui désigne une largeur de zone non zéro, et z qui désigne une largeur zone zéro.

CODAGE ENTROPIQUE :

Après la quantification scalaire des sous-bands de l'image, les indices $p(m, n)$ sont assignés à un flux de symboles, ainsi que nous le voyons dans le tableau 1, qui sont ensuite codés par le codage Huffman. Le tableau 1 indique comment les différents indices et zéro run-longueurs sont représentés par une série de 254 symboles. Les symboles 107 jusqu'à 254 sont utilisés pour transmettre les valeurs de l'indice entre -73 et 74. Si un index non nul est rencontré en dehors de cette gamme, le symbole d'échappement est ensuite transmis par la valeur entière de l'indice. Par exemple, un indice positif (resp. négatif) de valeur absolue inférieure à 256 et supérieure à 74, sera représenté par un symbole de 101 (resp. 102) et ensuite transmis par la valeur absolue de l'indice comme un 8-bit. De même, un indice de valeur absolue inférieure à 65536 et supérieur ou égal à 256, sera représenté par un symbole de 103 (resp. 104) et ensuite transmis par la valeur absolue de l'indice en tant que 16-bit. Il n'y a pas de symboles fournis pour transmettre un indice de valeur absolue supérieure ou égale à 65536.

Symboles 1 jusqu'à 100 sont utilisés pour transmettre la quantité de zéro.

Si la quantité de zéro est de plus de 100, on utilise le symbole 105 si elle est inférieure à 256, et on utilise le symbole 106 dans le cas où elle est supérieure à 256.

Les tables de codage Huffman sont indépendantes et donc l'image doit être contenue dans le format des données codées. La norme précise que les sous-images seront regroupées en 1 à 8 blocs pour le codage Huffman pour faciliter les capacités de transmission progressive. Toutes les sous-bands d'un bloc sont codées en utilisant la même table Huffman.



Figure. 4.27 : L'interface de comparaison entre l'image originale et image reconstruite

Après la réalisation de la compression selon le format FBI-Wavelet Scalar Quantization nous utilisons les facteurs de qualité (Taille de l'image originale, Taille de l'image reconstruite, Taux de compression, Erreur moyenne quadratique (MSE), Rapport signal sur bruit (SNR), Maximum du rapport signal sur bruit (SNR)) pour juger nos résultats

First-Compression est capable de réaliser une compression sur des images faciales. Pour donner un exemple, nous l'avons testé sur l'image de Lena. Voici les résultats avec les facteurs de qualité.

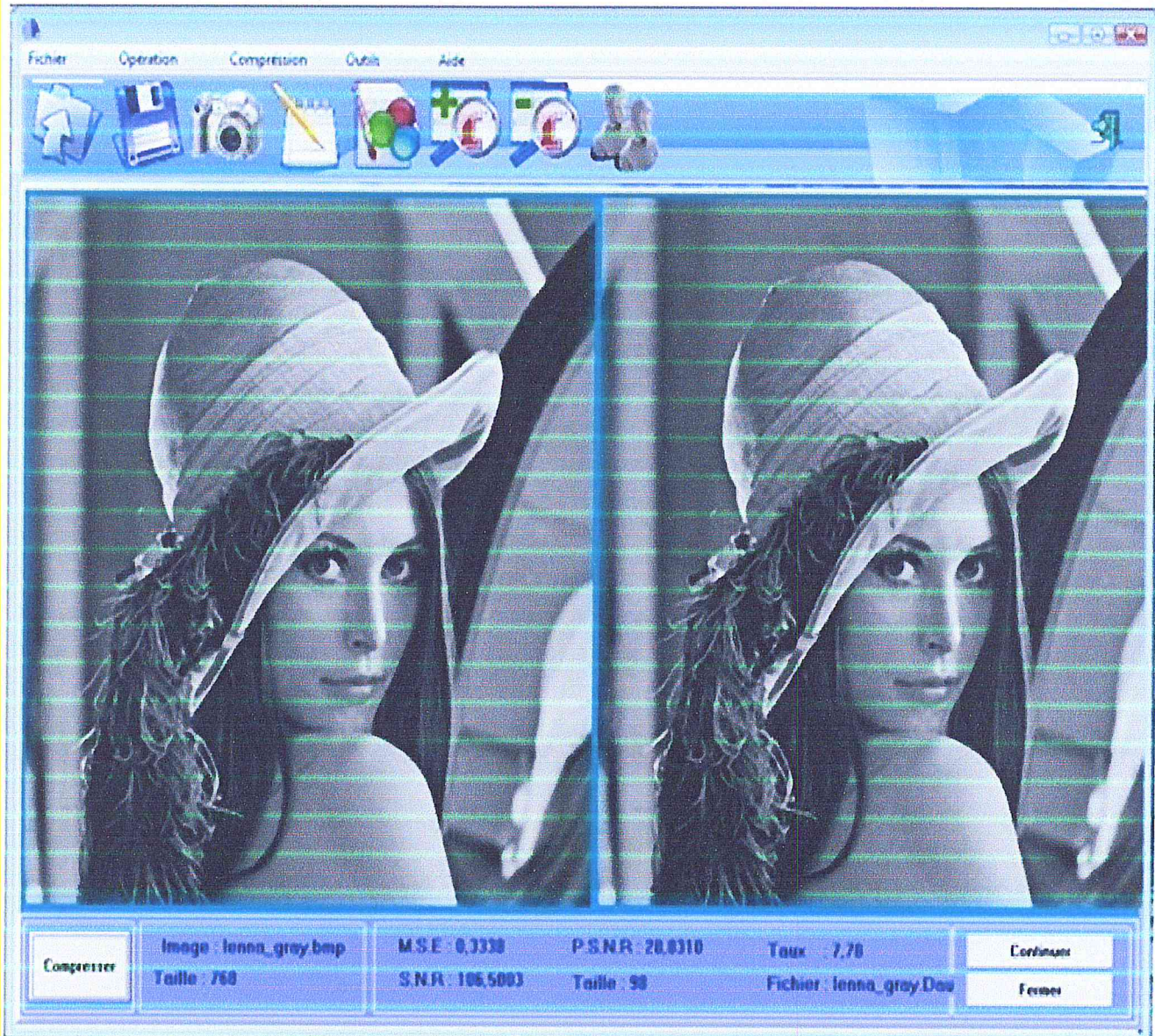


Figure 4.28 : L'interface de compression d'une image faciale

L'application permet de faire une comparaison sur la compression d'un ensemble d'images selon les facteurs de qualité (Taille de l'image originale, Taille de l'image reconstruite, Taux de compression, Erreur moyenne quadratique (MSE), Rapport signal sur bruit (SNR), Maximum du rapport signal sur bruit (PSNR)). Nous remarquerons que la taille de l'image Lena a diminué pour passer de 768 à 98 et son PSNR est de 28,83. Nous pouvons déduire que l'image est proche de l'original.

Pour améliorer ces résultats, notre application propose un autre type de filtre. Il s'agit du filtre moyen qui permet de réduire les grands décalages entre les pixels. Nous pouvons voir l'intérêt de ce filtre à travers la figure suivante.

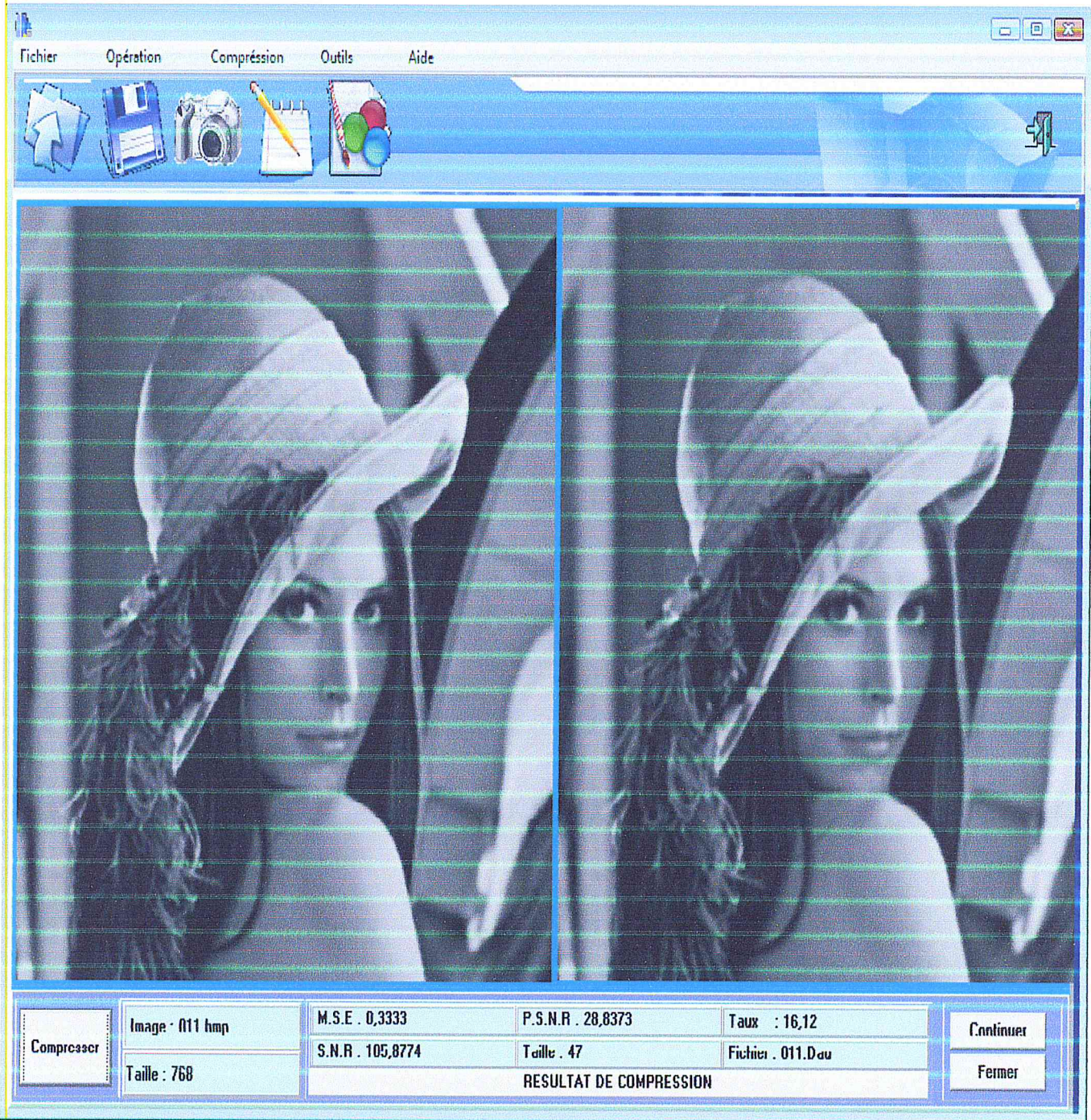


Figure 4.29 : L'intérêt du filtre moyen sur la compression d'une image faciale

L'application propose aussi une compression sur un dossier qui contient plusieurs images. En même temps, on mesure pour chaque image ses facteurs de qualité.

TABLEAU DE COMPARAISON

Image	Taille	Compression de Daubechis						Resultat de comparaison
		Taille	Taux	MSE	SNR	PSNR		
empreinte.bmp	2339	199	11,7538	235,7258	47,6753	0,7059		
face.bmp	1500	84	17,8571	597,8757	41,7472	8,5212		
fingerprnt.bmp	1597	156	10,2372	1067,1465	30,3019	19,9090		
img_facial.bmp	1079	98	11,0102	130,7974	46,1196	6,6761		
Lenna.bmp	768	133	5,7744	11,1201	79,0901	31,3251		
lenna_gray.bmp	768	98	7,0267	0,3338	106,5003	66,3058		
nanci.bmp	1500	87	17,2414	848,6823	38,3161	12,0242		
visage.bmp	305	47	6,4094	774,3430	43,0939	11,1075		
visage1.bmp	922	44	20,9545	1336,2491	20,0495	16,5636		



 Compare
  Quitter

Figure 4.30 : L'interface de compression d'un dossier

Les outils supplémentaires que nous avons implémenté dans notre application sont : Paint ,
Calculatrice, Bloc-notes.



Figure 4.31 : L'interface Bloc-notes

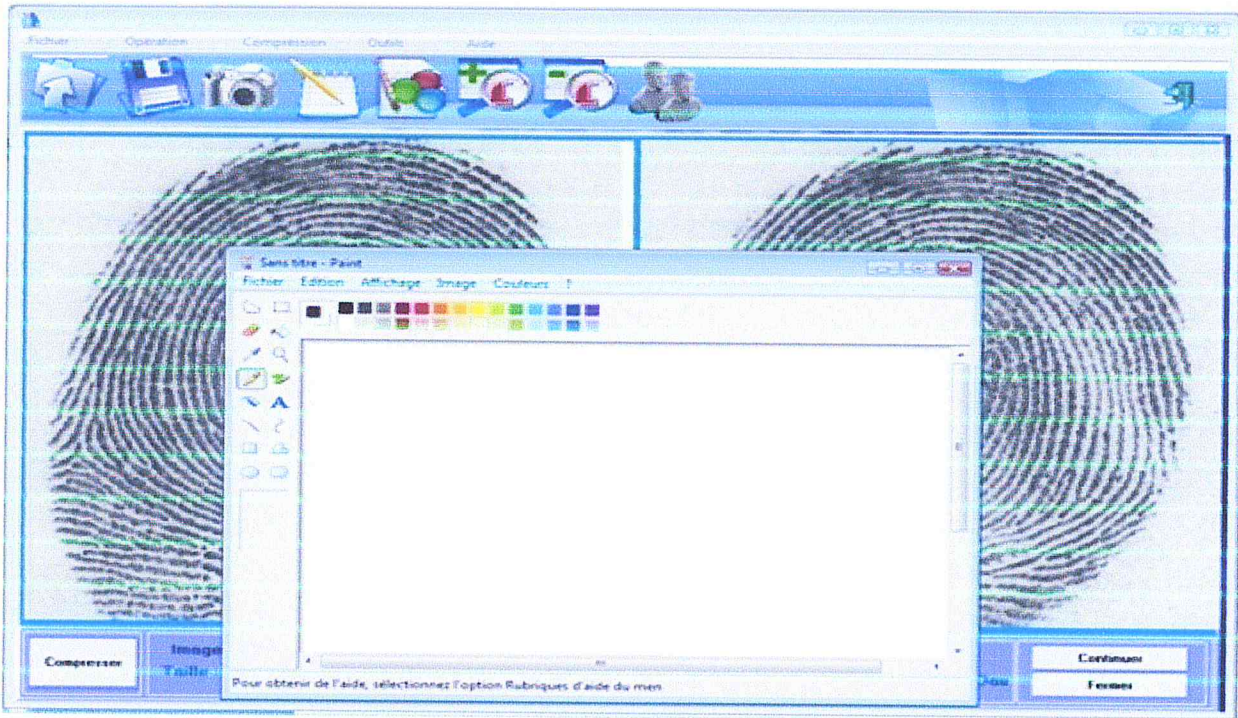


Figure 4.32 : L'interface Paint

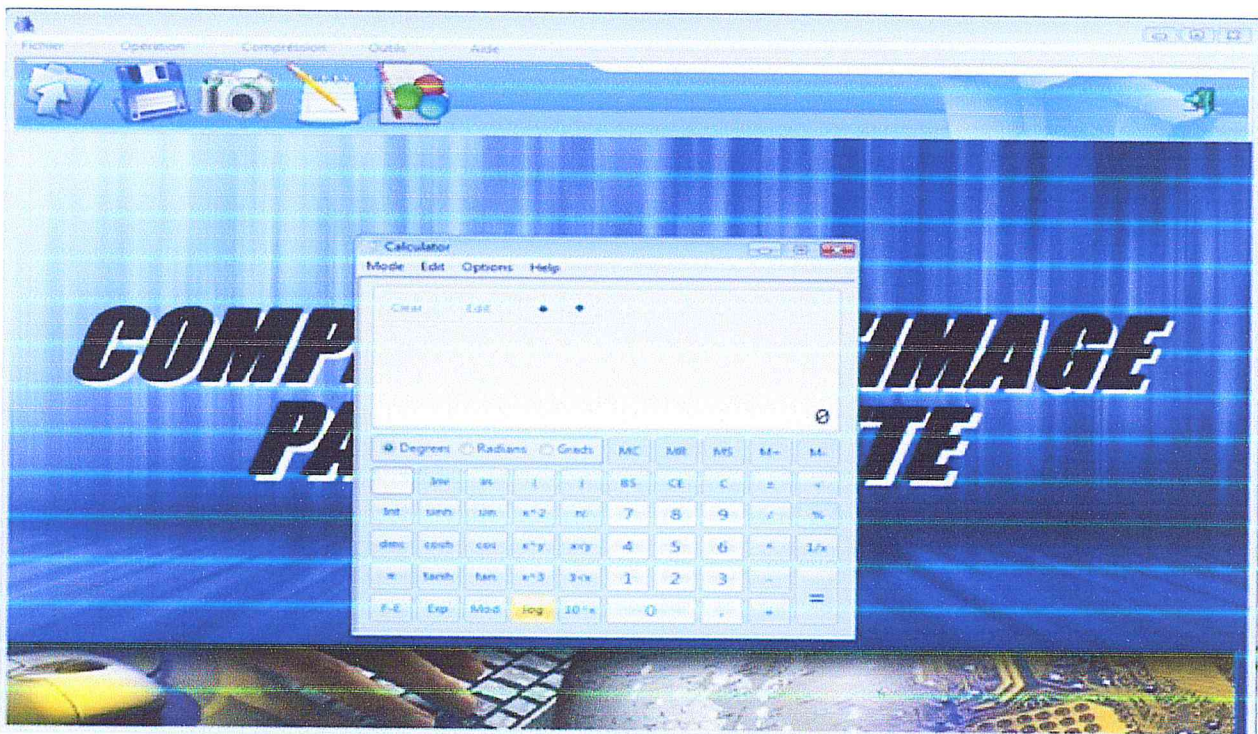


Figure 4.33 : L'interface Calculatrice

Pour faciliter la compréhension de notre travail nous avons présenté un fichier qui résume notre travail en termes de facteurs utilisés et des calculs obtenus.

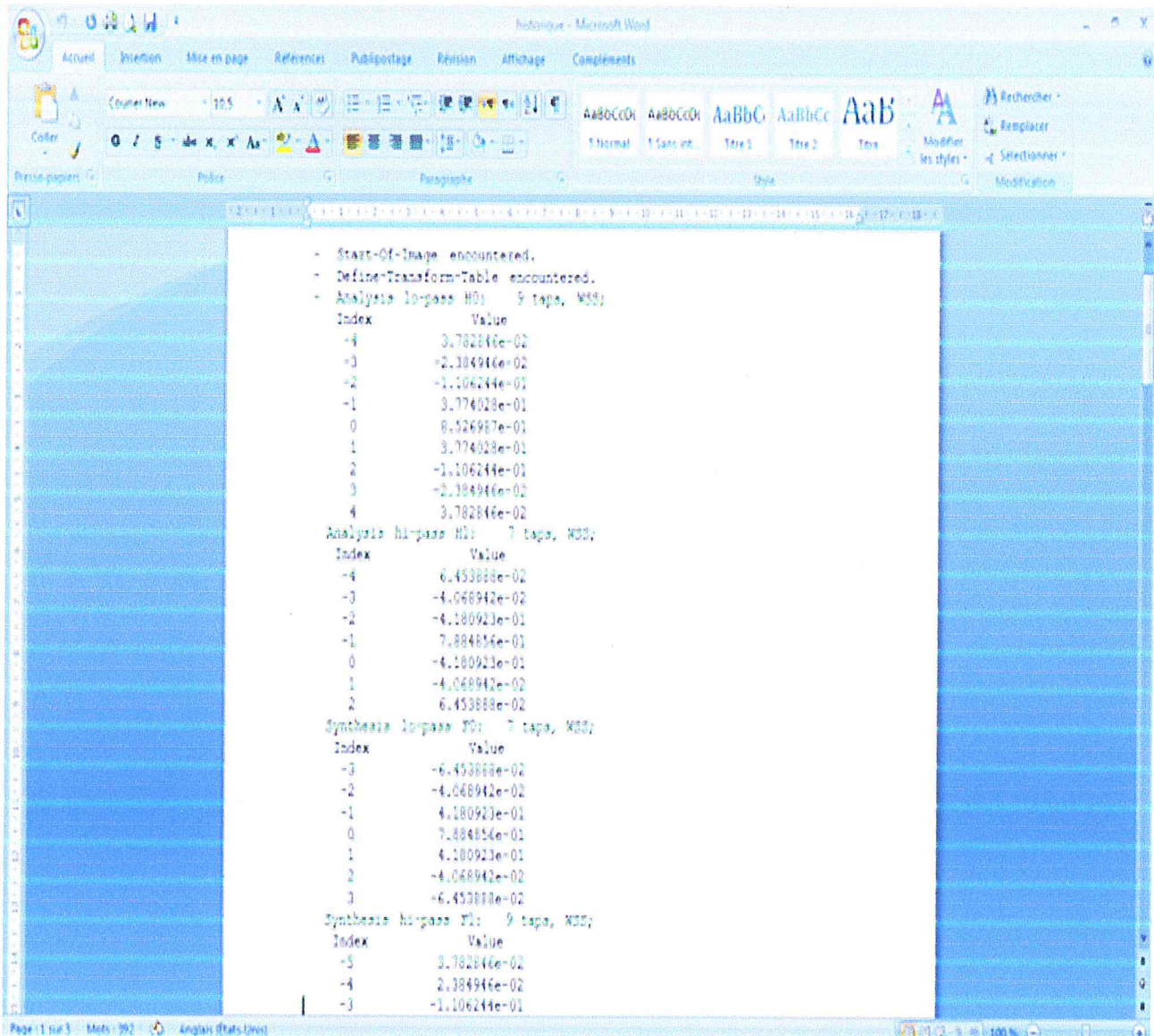


Figure 4.34: Représentation du fichier

La compression selon le format FBI WSQ utilise dans la phase de transformée en ondelette discrète (DWT) l'ondelette de type Daubechies Cohen Feauveau et pour visualiser l'intérêt de ce choix nous avons implémenté un autre type qui est l'ondelette de Haar.

Pour les ondelettes de Haar, les valeurs données par la fonction d'échelle correspondent à des valeurs moyennes sous échantillonnées des valeurs du vecteur initial. La différence de valeurs représentent les détails de l'image donnés par le filtre passe haut.

Soit deux mesures (l,r) . On note a la moyenne et d la différence de ces deux mesures.

Pour calculer la transformée en ondelettes de Haar d'une matrice de $n = 2^k$ données, il faut:

- Calculer la moyenne de chaque paire de données ($n / 2$ moyennes). $a = (l + r) / 2 =$ la fonction d'échelle
- Calculer la différence entre chaque donnée et sa moyenne ($n / 2$ différences). $d = l - a = a - r = (l - r) / 2 =$ la fonction d'ondelette
- placer les moyennes dans la première moitié de la matrice
- placer les différences dans la seconde moitié de la matrice;
- répéter le processus sur la première moitié des données (c'est pourquoi le nombre de données n doit être une puissance de 2).

La transformée inverse s'exprime par le calcul suivant:

- $l = a + d;$
- $r = a - d;$

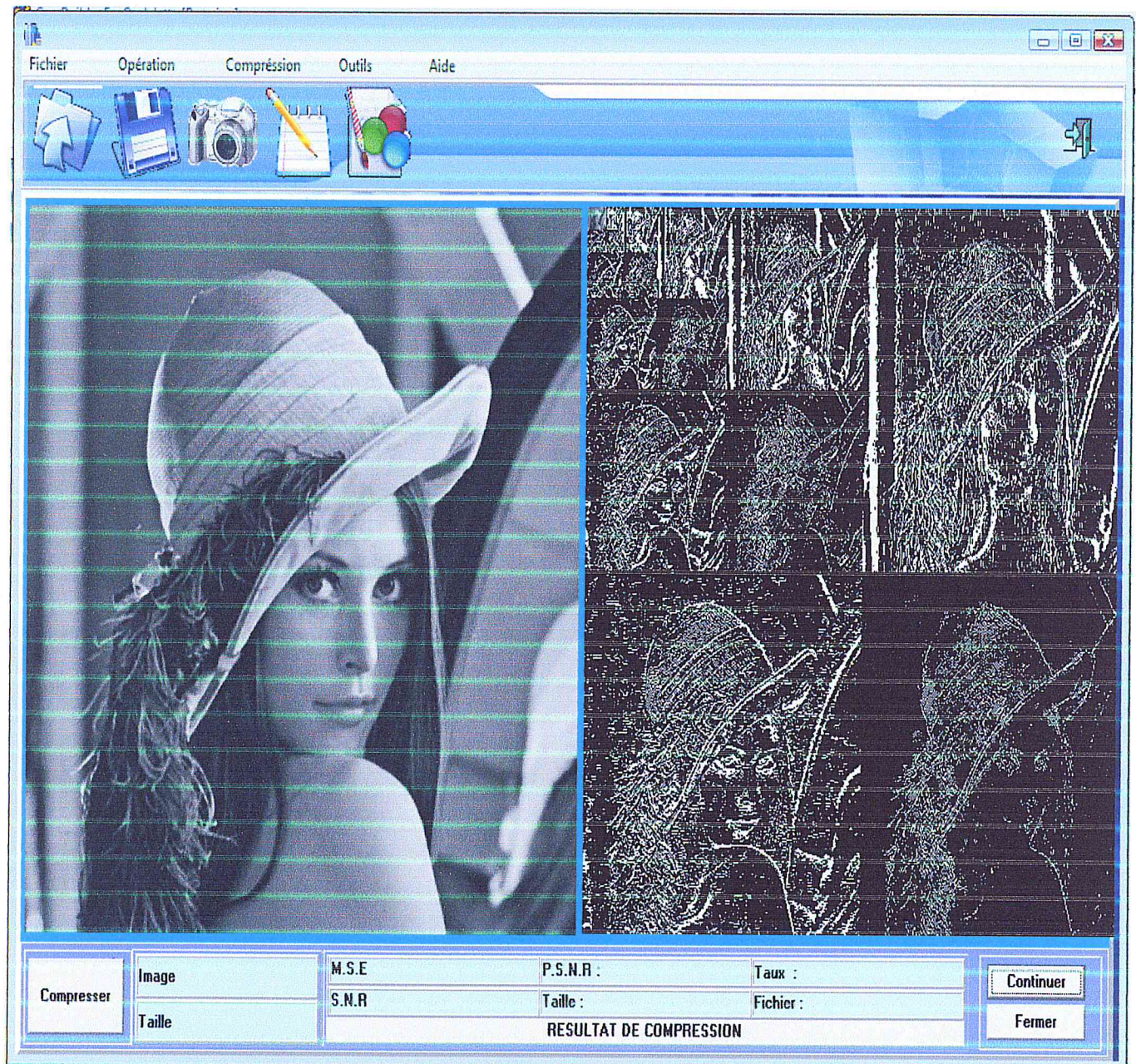


Figure 4.35: Décomposition en Ondelettes de Haar selon Mallat

La quantification

Après une transformation avec des ondelettes, nous nous retrouvons avec une matrice de coefficients correspondant à différents niveaux de détails. Suite à des constats lors d'études psychovisuelles, nous savons que les hautes fréquences sont moins importantes que les basses fréquences. Dans notre application, la quantification est restée simple : quantification linéaire revenant à diviser les coefficients par une constante choisit par l'utilisateur (voir figure 4.36) et à arrondir le résultat.

Voici la formule utilisée pour la quantification : $q = \text{sign}(v) \cdot (\text{int}) (|y| / \Delta b)$

q : est la valeur quantifiée.

y : la valeur à quantifier.

Δb : le pas de quantification.

Pour la dé-quantification, on utilise cette formule :

- $z = (q + r \cdot \text{sign}(q)) \cdot \Delta b$

r est une valeur fixée arbitrairement selon chacun, mais elle est souvent proche de 0,5.

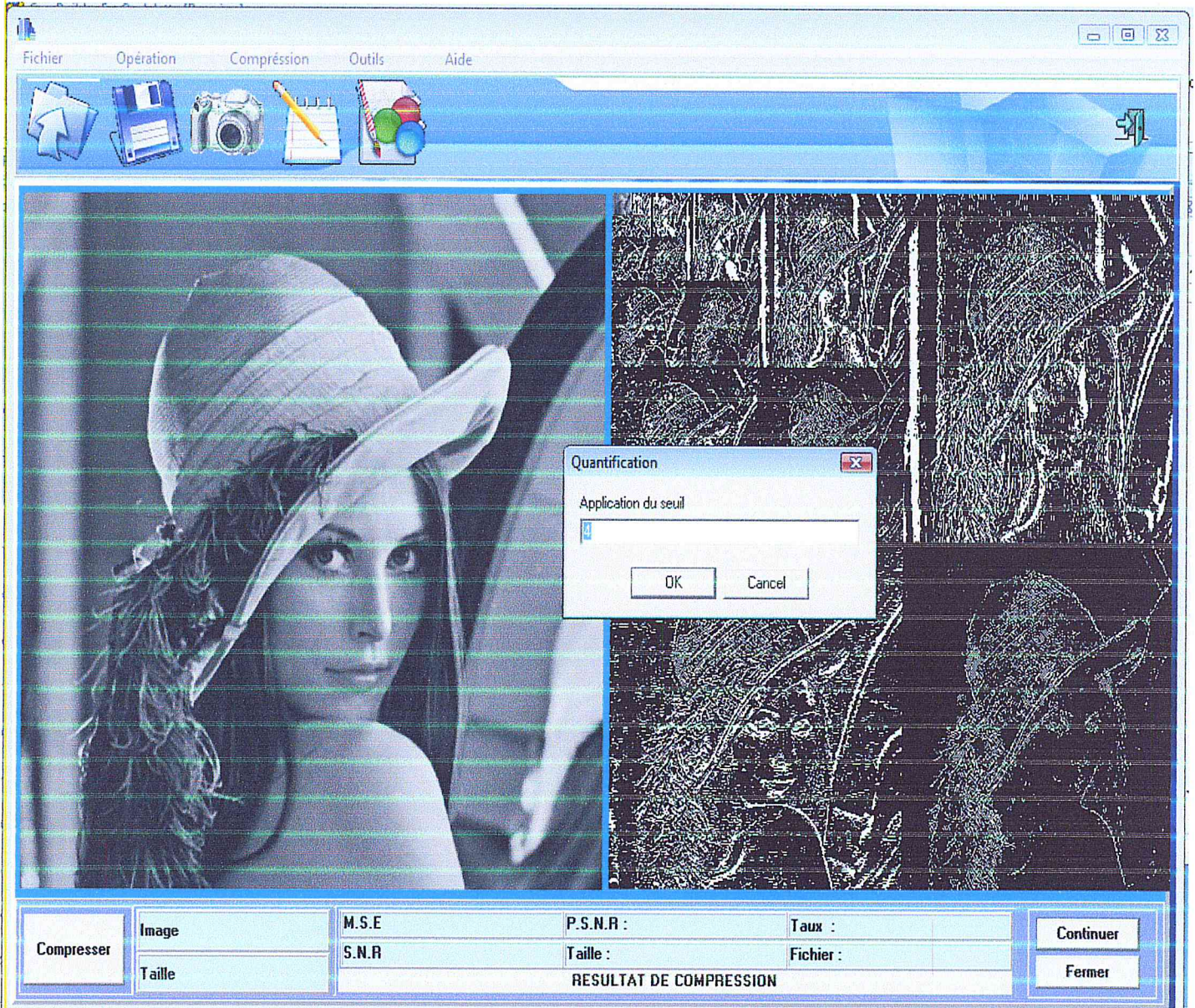


Figure 4.36: Quantification par l'ondelette de Haar

Le codage et le décodage sont simples, ils correspondent au codage Huffman utilisé par l'ondelette 9/7.

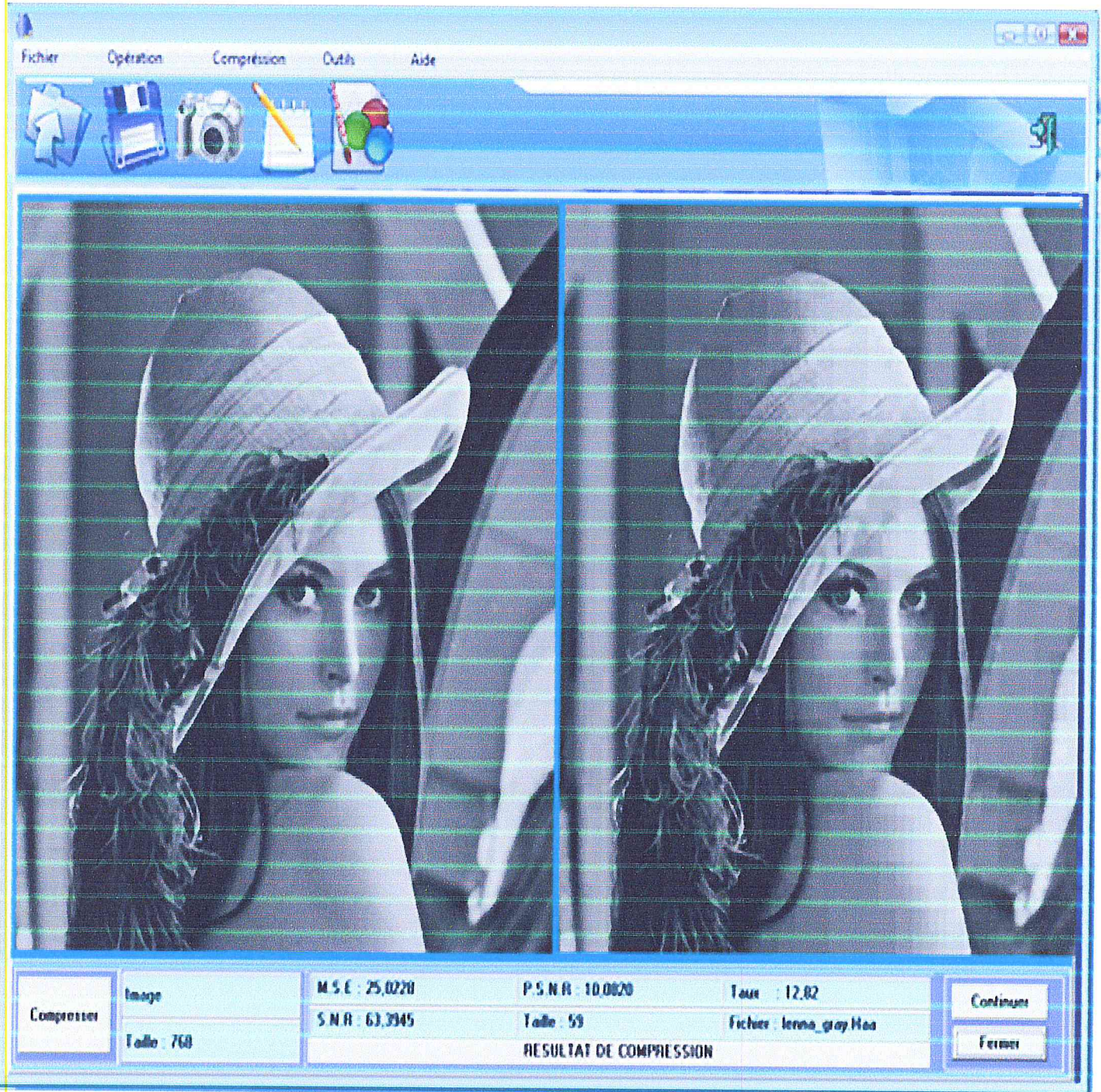


Figure 4.37 : Compression par l'ondelette Haar

La figure 4.37 représente une image de Lena compressée par l'ondelette de Haar. Par l'observation, on peut constater que cette méthode contient beaucoup d'artéfact et qu'elle est de faible qualité, comme le montrent également les calculs tels que le PSNR,

CONCLUSION

La technique de compression et de décompression des images des empreintes digitales que nous avons décrites dans cette étude a été appliquée à un ensemble d'images. La compression d'une image à l'aide de l'algorithme WSQ a permis de réduire notablement le volume des données tout en gardant leurs propriétés initiales. Dans la plupart des cas, elle a permis d'atteindre de bons résultats en termes de qualité des images traitées et de taux de compression. Ainsi, nous avons trouvé que l'erreur de distorsion obtenue pour différents taux de compression était plus petite avec notre méthode que dans le cas de la méthode de JPEG et des techniques basées sur les fractales. Rappelons que, pour atteindre de telles performances, nous avons utilisé des filtres de Cohen, Daubechies et Feveau 9/7 pour décomposer les images en sous-bandes.

Ainsi, le test de notre méthode sur l'image de Lena pour divers taux de compression, montre que les constituants de l'image décompressée sont bien représentés dans leur ensemble et qu'aucun artéfact n'est visible. Nous pouvons donc estimer que la majorité des objectifs fixés au début ont été atteints sauf quelques points qui doivent être pris en considération.

Notre outil ne nécessite aucun téléchargement ni aucune expérience dans son utilisation. Il est capable de réaliser les différentes étapes de coder et décoder FBI – WSQ et d'obtenir une image reconstruite de faible taille.

Avec les représentations graphiques des contenus, la navigation dans notre interface est devenue très simple à comprendre et facile à mettre en œuvre, même par un débutant.

Certains buts de notre application n'ont toutefois pas été atteints. Par exemple, elle ne peut traiter que les images de résolution 512 X 512 ainsi que les images de type bitmap. Par conséquent, le domaine de sécurité dans le cas du transfert des images reste à améliorer.

BIBLIOGRAPHIE

- [1] A.K. Jain, L. Hong and S. Pankanti, "Biometrics: Promising Frontiers for Emerging Identification Market", *Comm. ACM*, pp. 91-98, February. 2000.
- [2] J.-L. Dugelay et al., Recent Advances in Biometric Person Authentication , *IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, Orlando, Florida, May 2002.
- [3] International Biometric Group, homepage: www.biometricgroup.com
- [4] FBI, Federal Bureau of Investigation, homepage: www.fbi.gov
- [5] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, Face recognition: A literature survey , *ACM Computing Surveys (CSUR)*, Volume 35, Issue 4, December 2003.
- [6] W.A. Barrett, A survey of face recognition algorithms and testing results , *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, pp. 301-305, 1997.
- [7] A.K. Jain, A. Ross, and S. Pankanti, A prototype hand geometry-based verification system , in *Proc. Of 2nd Int l Conf. on Audio- and Video-based Biometric Person Authentication*, pp. 166-171, March 1999.
- [8] G.O. Williams, Iris Recognition Technology , *IEEE Aerospace and Electronics Systems Magazine*, Volume 12, Issue 4, pp. 23 -29, April 1997.
- [9] R.P. Wildes, Iris Recognition: An Emerging Biometric Technology , *Proceedings of the IEEE*, Volume 85, Issue 9, pp. 1348 -1363, Sept. 1997.
- [10] S.J. Vaughan-Nichols; Voice authentication speaks to the marketplace , *Computer*, Volume: 37 , Issue 3, pp. 13-15, March 2004.
- [11] B.H. Juang and T. Chen, The past, present, and future of speech processing , *IEEE Signal Processing Magazine*, Volume: 15 Issue: 3, pp. 24-48, May 1998.
- [12] L.L. Lee, T. Berger, and E. Aviczer, Reliable On-Line Human Signature Verification Systems , *IEEE Trans. on PAMI*, Vol. 18, No. 6, pp.643-647, June 1996.
- [13] W.J. Babler, Embryologic Development of Epidermal Ridges and Their Configurations, *Dermatoglyphics: Science in transition. Birth defects*, New York, Wiley-Liss, pp. 95-112, 1991.
- [14] A.K. Jain, S. Prabhakar and S. Pankanti, "Twin Test: On Discriminability of Fingerprints", *Proc. 3rd International Conference on Audio- and Video-Based Person Authentication*, , pp. 211-216, Sweden, June 6-8, 2001.
- [15] H. Ailisto and M. Linholm, "A review of fingerprint image enhancement methods", *International Journal of Image and Graphics*, Vol. 3, No. 3, pp. 401-424, 2003.
- [16] N. Yager and A. Amin, "Fingerprint verification based on minutiae features: a review", *Pattern Analysis and Applications*, Vol. 7, No. 1, pp. 94-113, April 2004.
- [17] D. Maio and D. Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 27-40, 1997.
- [18] Z. Bian, D. Zhang and W. Shu, "Knowledge-based fingerprint post-processing", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16, No. 1, pp. 53-67, 2002.
- [19] J.D. Stosz and L.A. Alyea, "Automated system for fingerprint authentication using pores and ridge structure", *Proceedings of SPIE in Automatic Systems for the Identification and Inspection of Humans*, Vol. 2277, pp. 210-223, October 1994.
- [20] V. Soifer, V. Kotlyar, S. Khonina and R. Skidanov, "Fingerprint identification using the directions field", *ICPR proceedings*, Vol. 3, pp. 586-590, 1996.
- [21] L.C. Ern and G. Sulong, "Fingerprint Classification Approaches: An Overview", *International Symposium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia, 13-16 August, 2001.

BIBLIOGRAPHIE

- [22] S. Pankanti, S. Prabhakar and, A.K. Jain, On the Individuality of Fingerprints , *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 8, pp. 1010-1025, August 2002.
- [23] D.P. Mital and E.K. Teoh, "An automated Matching Technique for Fingerprint Identification", *Proceedings of 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, Vol.2, pp. 806-911, 1996.
- [24] Jain, L. Hong and R. Bolle, On-Line Fingerprint Verification , *IEEE Transactions on PAMI*, Vol. 19, No. 4, pp. 302-314, 1997.
- [25] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric Recognition: Security and Privacy Concerns", *IEEE Security & Privacy Magazine*, Vol. 1, No. 2, pp. 33-42, March-April 2003.
- [26] L. O Gorman, Comparing passwords, tokens, and biometrics for user authentication , *Proceedings of the IEEE*, Vol. 91, No. 12, pp. 2019-40, Dec. 2003.
- [27] T. Matsumoto, Impact of Artificial "Gummy" Fingers on Fingerprint Systems , *Proceedings of SPIE*, Vol. 4677, pp. 275-289, 24-25 January 2002
- [28] M.L. Johnson, Biometrics and the threat to civil liberties , *Computer*, Volume: 37 , Issue: 4, pp. 90-92, April 2004.
- [29] Christian Cabal, "les méthodes scientifiques d'identification des personnes à partir des données biométriques et les techniques mises en uvre", *Office Parlementaire d'Evaluation des Choix Scientifiques et Technologiques*, juin 2003.
- [30] CNIL, Commission Nationale de l Informatique et des Libertés, www.cnil.fr.
- [31] Richard E .Woods Rafael C .Gonzalez . Digital Image Processing .Addison-Wesley Pub Co, ISBN : 0201180758, Paris, January 2002.
- [32] D. A. Huffman. A Method of the Construction of Minimum Redundancy Codes .In IRE, volume 40, pages 1098-1101, 1952.
- [33] Paul G.Howard and Jeffery Scott Vitter .Arithmetic Coding for Data Compression. Technical Report Technical report, DUKE-TR-1994-09, 1994.
- [34] B. Fong, G. Y. Hong, and A.C.M Fong .Constrained error propagation for efficient image transmission over noisy channels. *IEEE Transactions on Consumer Electronics*, 48(1):49-55, 2002.
- [35] A. Guyader, E. Fabre, and C. Guillemot. Joint source-channel turbo decoding of VLC encoded Markov sources. In GRETSI, pages CD-ROM, septembre 2001.
- [36] B.-J. Shieh, Y.-S. Lee, and C.-Y. Lee. A new approach of group-based VLC codec system with full table programmability. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2) :210_221, 2001.
- [37] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23 :337_343, 1977.
- [38] N. Moreau. Techniques de compression des signaux. Masson, Paris, 1995.
- [39] Kyung Sub Joo, D. R. Gschwind, and T. Bose. ADPCM encoding of images using a conjugate gradient based adaptive algorithm. In ICASSP - IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, volume 4, pages 1942-1945, 1996.
- [40] M. Babel, O. Déforges, and J. Ronsin. Décomposition pyramidale à redondance minimale pour compression d'images sans perte. In GRETSI, volume 1, page CD-ROM, 2003.
- [41] Z. Brahim and K.A. Saadi. Color image coding based on embedded wavelet zerotree and scalar quantization. In International Conference on Pattern Recognition, volume 1, pages 504-507, 2004.
- [42] David S. Taubman and Michael W. Marcellin. JPEG 2000 : Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

BIBLIOGRAPHIE

- [43] A. Gersho and R. M. Gray. Vector quantization and signal compression. Kluwer, Boston, 1992.
- [44] Guobin Shen, Bing Zeng, and M.-L. Liou. Adaptive vector quantization with codebook updating based on locality and history. *IEEE Transactions on Image Processing*, 12(3) :2003, 283-295.
- [45] R.A. Vander Kam and R.M. Gray. Lossy compression of clustered-dot halftones using sub-cell prediction. In *DCC - Proc. Data Compression Conference*, volume 1, pages 112_121, 1995.
- [46] S. Pateux. Segmentation spatio-temporelle et codage orienté-régions de séquences vidéo basés sur le formalisme MDL. PhD thesis, Université de Rennes 1, 1998.
- [47] T. Uto, M. Ikehara, and M. Okuda. Wavelet packet algorithm for quadtree-based embedded image coding. In *ICASSP - IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 4, pages 3517-3520, 2002.
- [48] D.R. Clewer, L.J. Luo, C.N. Canagarajah, D.R. Bull, and M.H. Barton. Efficient multiview image compression using quadtree disparity estimation. In *ISCAS - IEEE Int. Symp. on Circuits and Systems*, volume 5, pages 295_298, 2001.
- [49] S.S Maniccam and N.G.Bourbakis. Lossless image compression and encryption using SCAN. *Pattern Recognition*, 34(6) :1229-1245, 2001.
- [50] Y. Iano, F.S. da Silva, and A.L.M. Cruz. A fast and efficient hybrid fractal-wavelet image coder. *IEEE Transactions on Image Processing*, 15(1):98-105, 2006.
- [51] F. Davoine and S. Pateux. *Tatouage de documents audiovisuels numériques*. Hermès Science Publications, Lavoisier, France, 2003.
- [52] B.G. Monro, D.M. ; Sherlock. Optimal quantisation strategy for DCT image compression. In *IEE Proceedings Vision, Image and Signal Processing*, volume 143, pages 10-14, 1996.
- [53] M. J. Weinberg, G. Seroussi, and G. Sapiro. The LOCO-I Lossless Image Compression Algorithm : Principles and Standardization into JPEG-LS. *IEEE Trans. Image Processing*, 9 :1309_1324, 2000.
- [54] W. Sweldens. The lifting scheme : A costum-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2) :186_200, 1996.
- [55] Jonathan N. Bradley and Christopher M. Brislawn, The FBI Wavelet/Scalar Quantization Standard for gray-scale fingerprint image compression, Los Alamos National Laboratory, Los Alamos, NM 87545 Tom Hopper Federal Bureau of Investigation, Washington, D.C.20537.
- [56] Margaret A. Lepley, JPEG 2000 and WSQ Image Compression Interoperability, The MITRE Center for Integrated Intelligence Systems Bedford, Massachusetts 2001.
- [57] Digital Compression and Coding of Continuous- tone Still Images, Part 1, Requirements and Guide-lines, ISO Draft Int'l. Standard 10918-1,(a.k.a "the JPEG Standard"),American Nat'l. Standards Inst, Feb.1991.
- [58] C. M. Brislawn, "Classification of symmetric wavelet transforms" Tech. Rep. LA-UR-92-2823, Los Alamos Nat'l.Lab,Mar.1993. Revised draft.
- [59] WSQ Gray-Scale Fingerprint Image Compression Specification,(ver 2.0), Criminal Justice Information Services, Federal Bureau of Investigation, Feb.1993
- [60] G. Beylkin, R. Coifman, I. Daubechies, S. G. Mallat, Y. Meyer, L. Raphael et M. B. Ruskai. *Wavelets*. Jones and Bartlett, Boston, 1991.
- [61] I. Daubechies. Ten Lectures on Wavelets, tome 61 de CBMS-NSF Regional Conference Series in Applied Mathematics.Society for Industrial and Applied Mathematics, Philadelphia, 1992.
- [62] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbb{R})$. *Trans. Amer. Math. Soc.* 315(1):69-87, 1989.

BIBLIOGRAPHIE

- [63] S. G. Mallat. A theory for multiscale signal decomposition: The wavelet representation. *IEEE Trans. On Pattern and Machine Intelligence*, 11(7):674-693, 1989.
- [64] S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1997.