

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMARI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

D'Ingénieur d'Etat en Automatique

Thème

**Segmentation d'images par la
méthode JSEG et les ondelettes**

Proposé et dirigé par :
Mme HAMMOUCHE Sadia

Présenté par :
Mr DJENDER Samy
Mr GHEDDACHE Krim

Soutenu le : 21 / 10 /2010

Promotion 2010

Remerciements

Louange à DIEU qui nous a comblé de faveurs, et apporté son aide afin de réaliser le travail exposé tout le long de cette thèse.

Nous tenons à exprimer toute notre reconnaissance et notre gratitude à Mme HAMMOUCHE pour avoir accepté de nous encadrer, ses qualités humaines et pour tout ce que nous avons appris en travaillant avec elle.

Nos vifs remerciements vont aussi aux membres du jury qui ont accepté aimablement d'examiner et de juger notre travail.

Nous tenons également à remercier tous ceux qui ont contribué à la réalisation de ce mémoire.

Sommaire

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre 1 : Génialités sur le traitement d'images

1 Introduction.....	3
2 Définition de l'image et de l'image numérique.....	3
3 Différents types d'images numériques.....	4
3.1 Image binaire.....	4
3.2 Image intensité.....	5
3.3 Image couleur	6
4 Architecture générale d'un système de traitement d'images.....	7
4.1 Acquisition.....	7
4.2 Prétraitement et post-traitement.....	8
4.3 Le traitement numérique.....	8
4.4 Visualisation, transmission et stockage.....	9
5 Le filtrage	9
5.1 Les filtres linéaires	9
5.1.1 Filtres passe-bas (lissage).....	9
5.1.2 Filtres passe-haut (accentuation).....	10
5.1.3 Filtre Gaussien.....	10
5.2 Filtres non linéaires.....	11

5.2.1 Filtre médian.....	11
5.2.2 Filtre « le plus proche voisin radiométrique ».....	12
6 Définition de la segmentation d'image.....	12
7 Présentation de différents algorithmes de segmentation d'images.....	13
7.1 Approche basée sur les régions.....	14
7.1.1 Technique d'accroissement des régions (region-growing).....	14
7.1.2 Technique de partitionnement (splitting).....	14
7.1.3 Technique hybride de division/fusion (split and merge).....	16
7.2 Approche basée sur les contours.....	16
7.2.1 Opérateur gradient.....	17
7.2.2 L'approche Laplacienne.....	17
7.2.3 Opérateur de Sobel.....	18
7.2.4 Opérateur de Prewitt.....	18
7.3 Approche par classification de pixels.....	18
8 Définition de la texture	19
9 Analyse de la texture.....	20
10 Conclusion.....	21

Chapitre 2 : La méthode de segmentation des images texturées JSEG

1 Introduction.....	22
2 Principe de la segmentation JSEG.....	22
3 Quantification couleur (ou niveau de gris).....	23

3.1	Calcul des poids perceptuels.....	24
3.2	La quantification perceptuelle.....	27
4	Segmentation spatial.....	30
4.1	La mesure J.....	31
4.2	Détermination des J-images.....	33
4.3	Algorithme de segmentation.....	37
4.3.1	Détermination des germes.....	37
4.3.2	Croissance des germes.....	38
4.3.3	Fusion des régions.....	39
5	Conclusion.....	40

Chapitre 3 : Théorie sur les ondelettes

1	Introduction.....	41
2	La transformée de Fourier.....	41
3	Définition de la transformée en ondelettes.....	45
4	Transformée en ondelette continue.....	47
5	Transformée en ondelettes discrète.....	48
6	Analyse multirésolution.....	48
6.1	Analyse multirésolution à une dimension.....	49
6.1.1	Analyse mulirésolution et filtres numériques.....	49
6.1.2	Algorithme de transformation en coefficients d'ondelettes discrète et de reconstruction	51
a)	Algorithme d'analyse (décomposition)	51

b) Algorithme de reconstruction (la synthèse)	52
6.1.3 Illustration par un exemple numérique.....	53
6.2 Analyse multirésolution 2D.....	56
7 Conclusion.....	61

Chapitre 4 : Tests et Résultats

1 Introduction.....	62
2 méthode proposée.....	62
3 Influence du choix du niveau de décomposition de l'ondelette.....	66
4 Influence du choix du type de l'ondelette.....	69
5 Influence du seuil de fusion S.....	72
6 Influence du seuil de quantification QT.....	74
7 Influence de la taille de la fenêtre de balayage.....	76
8 Conclusion.....	79
Conclusion générale.....	80

Annexe

Bibliographie

Le traitement du signal a pour principal objet la description des signaux liés au monde réel dans un but de traitement, d'identification, de compression, de compréhension ou de transmission.

Le traitement d'image constitue l'une des grandes orientations du traitement de l'information acquise à partir de capteurs offrant des possibilités de plus en plus importantes. L'essor de potentialités des calculateurs a permis d'envisager des procédures de plus en plus complexes pour mettre en évidence ce que l'œil peut voir.

Chacun peut analyser l'image à sa manière, pour en dégager une impression et en extraire des informations précises. Dans le but de rendre cette opération possible, plus simple, plus efficace, plus agréable, et dans le but d'améliorer l'aspect visuel des images et d'en extraire les informations jugées pertinentes, le traitement d'image existe à cet effet. Le traitement d'image est donc l'ensemble des techniques et méthodes opérant sur l'image.

Dans ce travail, nous nous sommes intéressés à la partie segmentation d'images par classification de pixels et plus particulièrement à la méthode de segmentation appelée « JSEG ». C'est une méthode développée par Deng et Manjunath [4] en 2001 et qui a donné des résultats très satisfaisants.

La méthode JSEG, présente plusieurs étapes complexes, ce qui rend l'exécution du programme très lente surtout quand il s'agit d'images de tailles importantes. Pour remédier à ce problème, nous avons proposé de réduire la taille de l'image initiale en utilisant la transformée en ondelettes. L'information principale de l'image sera préservée alors que les détails seront éliminés. Par la suite, la segmentation par la méthode JSEG sera appliquée sur l'image réduite. A la fin, le résultat obtenu sur l'image réduite subira un agrandissement en utilisant la transformée en ondelettes inverse de même type.

Pour mener à bien notre travail, nous avons organisé notre mémoire en quatre chapitres.

Le premier chapitre est consacré aux notions générales sur le traitement d'images.

Dans le second chapitre nous avons présenté en détaille la méthode de segmentation « JSEG » en illustrant chaque étape par son algorithme et des images donnant les résultats intermédiaires.

La théorie des ondelettes est développée dans le troisième chapitre.

Le quatrième chapitre est consacré aux tests et résultats. L'influence d'un certain nombre de paramètres sur les résultats de la segmentation est étudiée dans le détail.

Au final, nous terminons notre travail par une conclusion générale.

Chapitre I

Génialités sur le traitement d'images

1. Introduction

Grâce aux développements technologiques réalisés dans le domaine du traitement d'images et avec la numérisation de celles-ci ; aujourd'hui l'intérêt des chercheurs est focalisé sur l'innovation de nouvelles techniques et la conception de nouveaux logiciels permettant la programmation des différentes opérations telles que la compression et la détection de contour afin de réaliser un temps de calcul optimal et une meilleure qualité du résultat.

Le traitement d'image désigne un ensemble de traitements automatisés qui permettent à partir d'images numérisées, de produire d'autres images numériques ou d'en extraire de l'information. La compréhension de ce domaine nécessitera quelques notions de base sur l'image et caractéristiques essentielles.

Nous aborderons dans ce chapitre, les notions générales de traitement d'images. Nous commencerons par définir une image et les notions qui s'y rattachent.

2. Définition de l'image et de l'image numérique

Le mot image renvoie à de nombreux sens. En liaison avec la vision, le mot désigne toute représentation d'un objet réel, qu'il s'agisse d'un tableau, dessin, ou d'une projection sur un écran.

Une image numérique est définie comme étant un signal fini bidimensionnel, échantillonné à valeurs quantifiées, dans un certain espace couleur ou niveau de gris. Elle est constituée de points appelés pixel (Picture element en anglais). Le pixel est l'unité de base d'une image numérique. Autrement dit, une image est une matrice $M \times N$ de valeurs entière prises sur un intervalle borné $[0, N_g]$, où N_g est la valeur maximale du niveau de gris.

Les principales caractéristiques d'une image numérique sont [8]:

- **La dimension** qui définit la taille de l'image et qui correspond au nombre total de pixels de l'image.

- **La résolution** qui correspond à la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production de l'image, elle est exprimée en nombre de pixels par unité de mesure.
- **La luminance** qui est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface.

- **Le contraste** qui est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 + L_2}{L_1 - L_2} \quad (I.1)$$

- **Les contours** qui représentent les frontières entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.
- **Le voisinage** qui est défini comme étant l'ensemble des pixels qui entourent dans toutes les directions un pixel donné.
- **L'histogramme** qui est un graphique statistique permettant de représenter la distribution des niveaux de gris des pixels d'une image.

3. Différents types d'images numériques

Principalement, on distingue trois types d'images numériques largement répandus et qui sont les images binaires, en niveau de gris et en couleur.

3.1. Image binaire

Une image binaire est une matrice rectangulaire dont les éléments valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc.

On peut transformer une image à plusieurs niveaux de gris en une image binaire. Cette opération est effectuée en comparant les valeurs des pixels par rapport à un seuil. Les valeurs supérieures à ce seuil prennent la valeur 255 (blanc) et celles inférieures prennent la valeur 0 (noir).

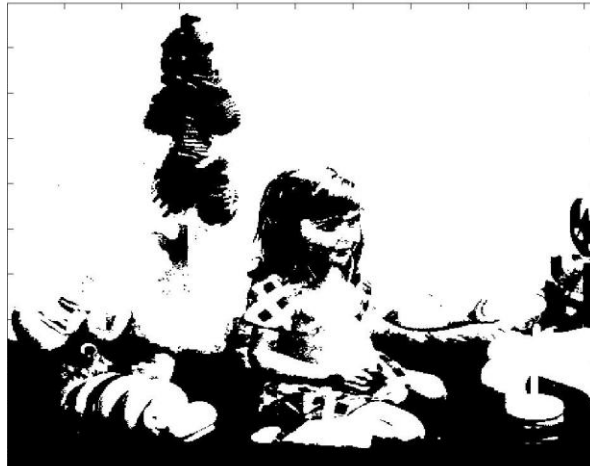


Fig .I.1: Exemple d'image binaire

3.2 Image intensité

Une image intensité ou image en niveau de gris est une matrice dans laquelle chaque élément est un entier compris entre 0 (noir) et 255 (blanc). On parle aussi d'image en niveau de gris, car les valeurs comprises entre 0 et 255 représentent les niveaux de gris. Par exemple 10 représente un gris foncé et 210 représenté un gris clair.



Fig.I.2: Exemple d'image intensité

3.3 Image couleur

L'œil humain analyse la couleur à l'aide de trois types de cellules photo réceptrices : les cônes. Ces cellules sont sensibles aux basses, moyennes, et hautes fréquences (rouge, vert et bleu). Pour représenter la couleur d'un pixel il faut donc donner trois nombres, qui correspondent au dosage des trois couleurs de base. La manière la plus naturelle de représenter une image consiste à utiliser trois matrices R, V, B (voir figure I.3). Toutefois pour les grandes images, ceci conduit à l'occupation d'un espace mémoire très important. Pour cette raison, une autre manière très économique de représenter les images couleur connu sous le nom de la représentation indexée, a vu le jour. Cette représentation ne permet de représenter qu'un nombre limité de couleurs. Ces couleurs sont mémorisées dans une table de couleurs dite colormap qui est une matrice $n \times 3$ où n est le nombre de couleurs. L'image est une matrice contenant des nombres entiers compris entre 1 et n , chaque entier jouant le rôle d'un index relatif à la table de couleurs. Par exemple, l'élément représenté sur la figure I.4 contient l'entier 18, ce qui ramène à la ligne 18 de la table des couleurs pour connaître la couleur du pixel correspondant.



Fig.I.3: Exemple d'image couleur

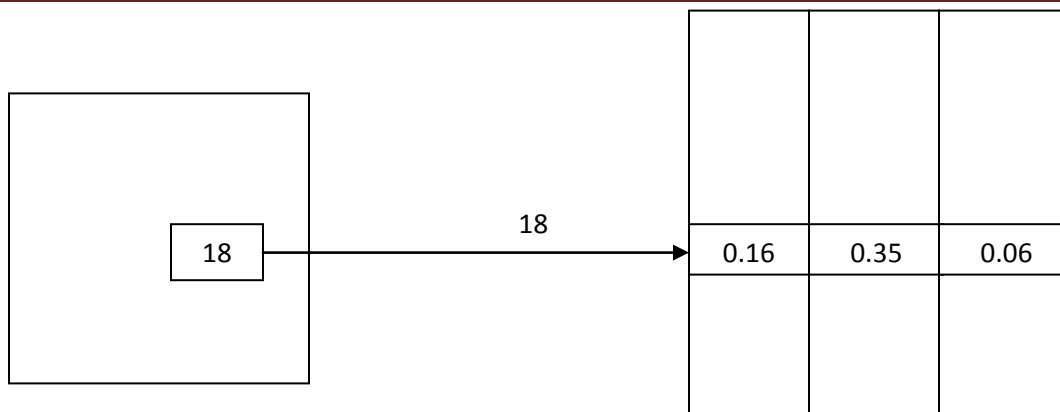


Fig.I.4 : Principe de l'image indexée [2]

4. Architecture générale d'un système de traitement d'images

Le traitement d'images désigne une discipline de l'informatique et des mathématiques appliquées, qui étudie les images numériques et leurs transformations dans le but d'améliorer leur qualité ou d'extraire des informations pertinentes.

Un système de traitement d'image est composé de plusieurs parties, la figure I.5 montre les différentes étapes de traitement :

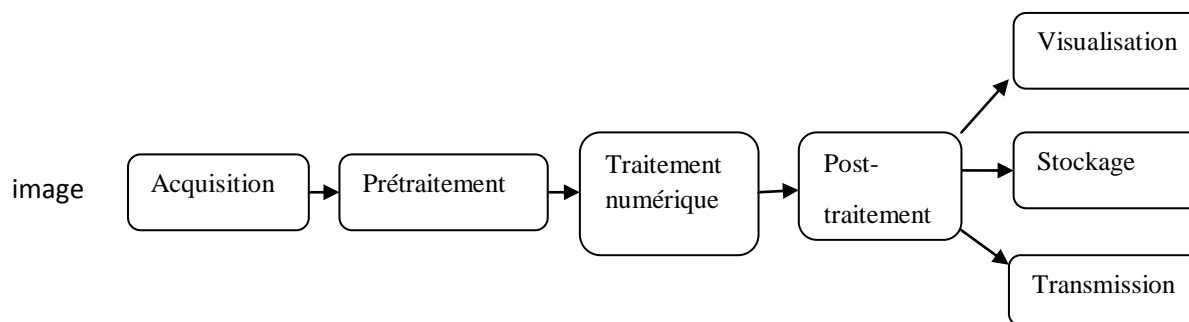


Fig. I.5: Architecture générale d'un système de traitement d'image

4.1 Acquisition

L'acquisition d'une image est l'opération qui permet le passage de l'information réelle à une représentation numérique. Elle est réalisée en 3 étapes : la transformation du signal optique en un signal analogique (électrique), l'échantillonnage qui consiste à multiplier le signal analogique par une série d'impulsion unité dans le but d'avoir des

échantillons du signal, et la quantification qui est une traduction des échantillons en valeurs numériques selon une règle de codage choisi.

4.2 Prétraitement et post-traitement

Le prétraitement concerne les images acquises tandis que le post-traitement concerne les images traitées. Dans les deux étapes, les images ne peuvent échapper aux effets de dégradation due essentiellement aux phénomènes physiques.

L'opération de prétraitement a pour but principal la correction de l'imperfection du système d'acquisition, alors que l'opération post-traitement vise surtout à mettre en forme pour éventuellement une visualisation, transmission ou un stockage.

Les techniques de prétraitement les plus utilisées reposent sur le filtrage numérique et la modification de l'histogramme.

4.3 Le traitement numérique

Le traitement numérique d'image est l'ensemble des méthodes qui permettent de décrire quantitativement le contenu d'une image.

On distingue deux types de traitement, les traitements de bas niveau et les traitements de haut niveau. Cette distinction est liée au contenu sémantique des entités traitées et extraites de l'image.

Les traitements de bas niveau opèrent en général sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire la liaison avec la réalité. Ils opèrent plutôt sur des données de nature numérique.

Les traitements de haut niveau s'appliquent à des entiers de nature symbolique associés à une représentation de la réalité extraite de l'image ils sont relatifs à l'interprétation.

4.4 Visualisation, transmission et stockage

La visualisation est une opération qui permet de transformer le signal numérique qui est la matrice image en un signal analogique visible par l'œil de l'observateur à travers un dispositif d'affichage.

L'image peut être transmise vers une station d'archivage ou de traitement qui peut être différente de la station d'acquisition.

Les images sont archivées sur des supports destinés à cet effet comme par exemple les bandes magnétiques, les disques durs,...

5. Le filtrage

L'amélioration des images est essentiellement obtenue par filtrage. Il consiste soit à atténuer le bruit, soit à accentuer les contours pour mieux extraire les informations voulues. Il existe un grand nombre de filtres classés principalement en deux catégories, les filtres linéaires et les filtres non linéaires.

5.1 Les filtres linéaires

Ils sont appelés ainsi car ils possèdent la propriété de linéarité du produit de convolution. Filtrer une image $I(x,y)$, consiste à la convoluer avec une fonction $F(x,y)$ qui est la réponse impulsionnelle du filtre.

Les filtres linéaires les plus utilisés, sont les filtres passe-bas, passe-haut et Gaussien.

5.1.1 Filtres passe-bas (lissage)

Ce filtre n'affecte pas les composantes basses fréquences dans les données d'une image, mais doit atténuer les composantes de hautes fréquences. L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image. Elle peut être répétée plusieurs fois, ce qui crée un effet de flou. En pratique, il faut choisir un compromis entre l'atténuation du bruit et la conservation des détails et contours significatifs. Pour réaliser ce filtrage, on effectue le produit de convolution de l'image par une fonction de voisinage définie d'une façon générale par :

$$\mathbf{H}_b = \left(\frac{1}{b+2}\right)^2 \begin{bmatrix} \mathbf{1} & -\mathbf{1} & -\mathbf{1} \\ \mathbf{b} & \mathbf{b}^2 & -\mathbf{1} \\ \mathbf{1} & \mathbf{b} & \mathbf{1} \end{bmatrix} \quad (\text{I.2})$$

Où $\left(\frac{1}{b+2}\right)^2$ est un facteur de normalisation

Les deux types de filtres passe-bas les plus utilisés sont pour $b=1$ et $b=2$

5.1.2 Filtres passe-haut (accentuation)

Le renforcement des contours et leur extraction se fait dans le domaine fréquentiel par l'application d'un filtre passe-haut. Le filtre passe-haut a les caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composantes hautes fréquences d'un signal, mais doit atténuer les composantes basses fréquences. Le filtrage passe-haut est réalisé par le produit de convolution de l'image par les masques suivants :

$$\mathbf{H}_1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -1 \\ 1 & -2 & 1 \end{bmatrix} \quad (\text{I.3})$$

5.1.3 Filtre Gaussien

C'est un filtre de lissage très utilisé pour sa facilité de mise en œuvre. L'expression de la Gaussienne en deux dimensions est donnée par :

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (\text{I.4})$$

Où X, Y représentent les coordonnées d'un pixel, σ est la variance. L'intérêt du filtrage Gaussien est que l'on peut régler très facilement le degré du filtrage à travers le paramètre σ . L'inconvénient de ce filtre est qu'il dégrade les contours en introduisant du flou.

5.2 Filtres non linéaires

Ils constituent une alternative au filtrage linéaire. La nouvelle valeur du pixel n'est plus une combinaison linéaire des valeurs des pixels voisins, mais une autre forme de calcul excluant toute relation linéaire entre le pixel et son voisinage.

Parmi ces filtres, on peut citer le filtre médian et le filtre du plus proche voisin radiométrique.

5.2.1 Filtre médian

Nous avons vu qu'un filtre passe-bas permet de réduire le bruit haute fréquence.

Toutefois, l'inconvénient de ce filtre est une impression de flou sur l'image débruitée. Il existe un filtre qui permet de réduire certains types de bruit en dégradant très peu les contours, il s'agit du filtre médian. Ce filtre est particulièrement efficace pour les bruits de type poivre et sel c'est-à-dire les bruits qui affectent seulement un petit nombre de pixels, mais en modifiant très fortement l'intensité des pixels touchés.

Le filtre médian affecte à un pixel la valeur médiane des intensités dans son voisinage. Pour expliquer le fonctionnement du filtre médian, prenons un exemple avec un voisinage de 3*3 pixels.

10	0	40
255	10	150
230	20	100

Image avant filtrage

10	0	40
255	40	150
230	20	100

Image après filtrage

Fig.I.6: Exemple d'application du filtre médian

Le filtre médian range par ordre croissant les intensités du voisinage. On obtient alors

0 10 10 20 40 100 150 230 255

Il affecte ensuite au pixel central qui valait initialement 10 l'intensité qui se trouve au milieu du rangement ci-dessus, c'est-à-dire 40.

5.2.2 Filtre « le plus proche voisin radiométrique »

Par ce filtrage, le pixel central est remplacé par la valeur moyenne des K pixels voisins dont les valeurs radiométriques (niveau de gris) sont proches de la sienne.

10	0	40
255	10	150
230	20	100

Image avant filtrage

10	0	40
255	35	150
230	20	100

Image après filtrage

Fig. I.7: Exemple d'application du filtre le plus proche voisin radiométrique

Le pixel central qui a la valeur radiométrique 10 sera remplacé par la valeur 35 qui est la moyenne des valeurs 10, 20 et 40 qui sont les valeurs les plus proches de 10. Dans ce cas K est pris égal à 3.

6. Définition de la segmentation d'image

De plus en plus, les images sont omniprésentes dans les systèmes d'information et devant cette quantité, il faut être capable de traiter les informations qu'elles contiennent. Pour cela, il faudrait pouvoir distinguer le fond de la scène des objets qu'elle contient mais également les objets entre eux et c'est là qu'intervient le processus de segmentation. En effet, la segmentation d'image est une étape de traitement de l'image permettant de manière générale de regrouper les pixels d'une image ayant des propriétés (couleurs, intensité lumineuse, position,...) communes en classes. Plus formellement, la segmentation correspond au partitionnement d'une image R en N régions R_i (ensemble de pixels) disjointes et homogènes selon un critère ou un prédicat bien déterminé. C'est-à-dire:

$$- \bigcup_{i=1}^N R_i = I, \quad i = 1, \dots, N. \quad (\text{I.5})$$

$$- R_i \cap R_j = \emptyset \quad \forall i \neq j \quad (\text{I.6})$$

$$- R_i \text{ est connexe } \quad \forall i \in \{1, \dots, N\} \quad (\text{I.7})$$

$$- P(R_i) = \text{vrai} \quad \forall i \in \{1, \dots, N\} \quad (\text{I.8})$$

$$-P(R_i \cup R_j) = \text{faux} \quad \forall i \neq j \text{ et } R_i \text{ adjacence à } R_j \quad (I.9)$$

L'étape de segmentation est nécessaire afin de reconnaître, isoler et traiter les informations utiles contenues dans une scène. Cependant, s'il est facile pour une machine de rechercher des informations dans un texte, le problème en termes d'image est beaucoup plus délicat du fait de leur nature. Si un être humain peut aisément distinguer les différentes zones composant une image du fait des mécanismes de haut niveau mis en jeu par notre cerveau, de notre expérience, de notre façon de prévoir les objets..., pour un ordinateur, la tâche est beaucoup plus difficile. En effet, une image contient des objets hétérogènes à plusieurs niveaux et pour segmenter correctement une image, il faudrait prendre en compte les composantes telles que la couleur, la luminosité, les textures et toutes sortes d'autres critères qui sont propres à chaque image.

Ainsi, le problème de la segmentation est un sujet complexe à aborder car il n'existe pas de technique universelle ou critère unique permettant de donner des résultats acceptables. C'est pour cela que certaines techniques qui seront bien adaptées dans certains cas le seront beaucoup moins dans d'autres. On ne peut donc dire d'un algorithme de segmentation s'il est bon ou mauvais mais il faut simplement bien étudier la finalité de la segmentation afin d'utiliser le programme adéquat.

Selon le domaine d'application (imagerie médicale, robotique, recherche d'information...), on utilisera certains algorithmes plutôt que d'autres en fonction du résultat recherché. Par exemple, dans certains domaines, tels que la robotique, il est nécessaire de privilégier la vitesse de segmentation à la précision. Il faut donc trouver les algorithmes qui réalisent un compromis entre vitesse et précision.

7. Présentation de différents algorithmes de segmentation d'images

Comme nous l'avons déjà dit précédemment, il existe plusieurs approches au problème de segmentation d'image. En général ces approches peuvent être classées en trois grandes familles :

- approche basée sur les régions,

- approche basée sur les contours et
- approche basée sur la classification des pixels.

7.1 Approche basée sur les régions

Ce type d'algorithmes repose sur le fait qu'une image est composée de régions et que l'on va pouvoir segmenter une image en travaillant sur ces régions. On va donc, en fonction des critères définis, modifier la taille des régions en agrandissant ou en diminuant leurs taille jusqu'à ce qu'une certaine condition soit satisfaite (nombre de régions, temps, ...).

7.1.1 Technique d'accroissement des régions (region-growing)

C'est une technique très répandue et qui a démontré sa robustesse. Son principe est simple: on choisit arbitrairement un pixel qui servira de « graine » que l'on compare à ses voisins, la région grandit à partir de la graine en ajoutant les pixels voisins similaires selon certains critères ; quand une région cesse de grandir, on prend un autre pixel «graine» n'appartenant à aucune région et on recommence le processus. On s'arrête lorsque tous les pixels appartiennent à une région.



Fig. I.8: Exemple illustrant la technique d'accroissement de régions

7.1.2 Technique de partitionnement (splitting)

A l'inverse de la technique de l'accroissement de région, on va considérer ici l'image toute entière pour ensuite la «diviser» en un ensemble de régions cohérentes par

partitionnement des régions existantes. On estime que l'image est au départ constituée d'une seule région correspondant à l'image toute entière. On étudie les pixels contenu dans une région et on regarde si tous les pixels vérifient un ensemble de contraintes de similarité. Si c'est le cas, alors la région courante correspond à une région de l'image sinon, on divise la région en sous-régions (classiquement en quatre sous-zones égales) et on recommence le processus sur celles-ci. Le processus continue jusqu'à ce que l'on ne puisse plus partitionner.

L'inconvénient de cette méthode est de partitionner l'image en régions qui n'ont pas de sens sémantique au niveau des objets qu'elle contient. La segmentation est souvent trop fine et l'image est souvent divisée en de nombreuses petites zones qui ont, comme nous l'avons dit, peu de sens. De plus, selon le découpage choisi, souvent la forme des zones ne correspond pas aux contours des objets. Un algorithme classique illustrant cette méthode est le Quadtree.

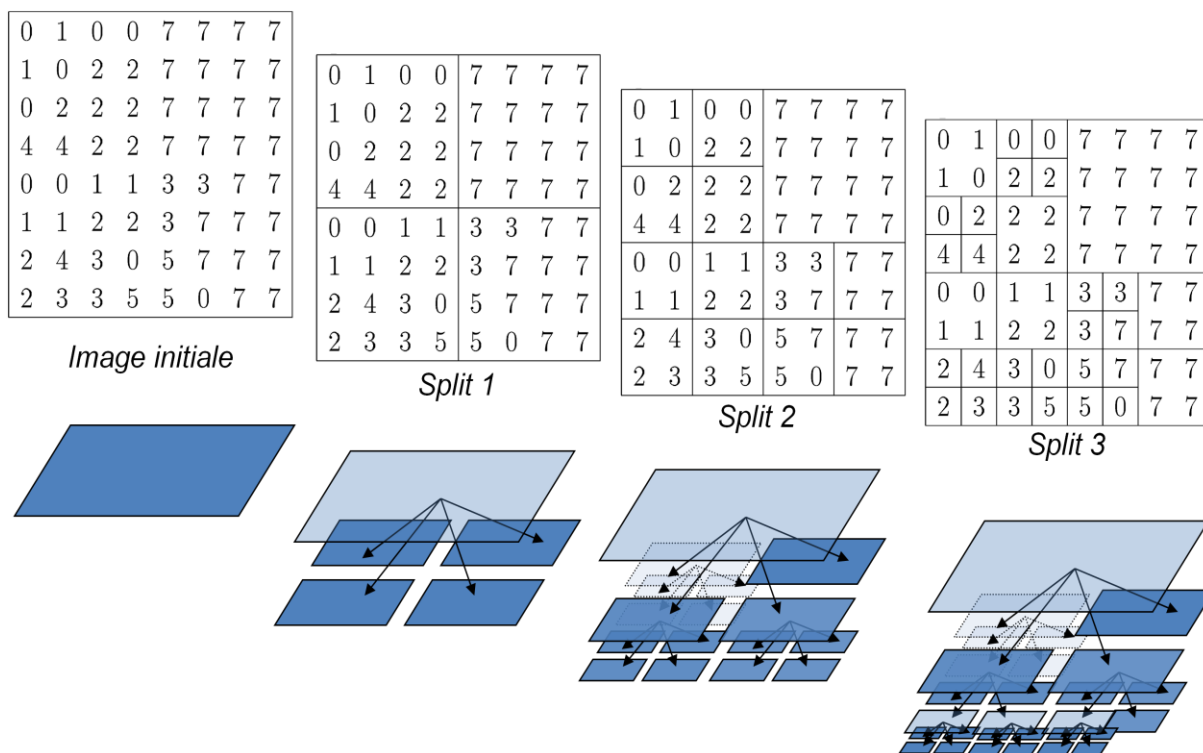


Fig. I.9: illustration de la technique de partitionnement

7.1.3 Technique hybride de division/fusion (split and merge)

Pour remédier à certains problèmes de la technique de partitionnement, une méthode a été mise au point permettant de retravailler une image après qu'elle ait été partitionnée par la méthode précédemment décrite. On partitionne l'image par la technique de splitting où chaque région est comparée à ses voisines et on regarde si elles vérifient des contraintes de similarité. Si c'est le cas, on fusionne les deux régions. Sinon, on passe à la région suivante.

Cette technique combine les deux méthodes précédentes et offre des résultats satisfaisants.

A titre d'exemple (voir figure I.10), on considère une image I et on suppose qu'elle ne vérifie pas un certain critère d'homogénéité, on la divise alors en quatre sous parties : I_1 , I_2 , I_3 et I_4 . Les pixels de I_1 , I_2 et I_3 vérifient le critère mais pas ceux de I_4 que l'on divise de nouveau. On commence le processus de fusion et on voit que I_{43} et I_{44} ont des similarités alors on les fusionne.

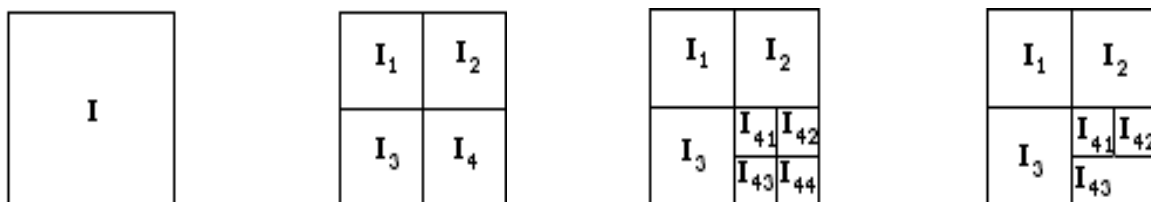


Fig. I.10: Technique de division-fusion

7.2 Approche basée sur les contours

Une autre approche, complètement différente de la précédente, consiste à essayer de déterminer les contours des différents objets, personnages... d'une scène. Ainsi, a priori, chaque contour peut définir une zone représentant un objet. En général, un contour se caractérise par un changement brutal de couleur ou d'intensité. De ce fait on peut détecter ces contours par des méthodes dérivatives. Il existe plusieurs manières de détecter les contours dans une image parmi elles, on cite : l'approche gradient, l'approche Laplacienne et l'utilisation des opérateurs de Sobel, Prewitt, Roberts etc.

Le gradient est très utilisé pour détecter les contours. La méthode est très simple ; on commence par le calcul de la norme du gradient pour chaque pixel de l'image, les pixels où le gradient est maximum correspondent à une frontière. En général, le gradient n'est pas utilisé seul, on procède par la suite à une binarisation de l'image gradient par un algorithme de seuillage pour obtenir une image binaire avec des contours propres

7.2.1 Opérateur gradient

Le gradient en un point est défini par :

$$\nabla I(x, y) = \left[\frac{dI}{dx} \vec{u} + \frac{dI}{dy} \vec{v} \right] = G_x \vec{u} + G_y \vec{v} \quad (\text{I.10})$$

Sous forme numérique, ce vecteur sera :

$$\nabla I(x, y) = [I(x, y) - I(x-1, y)] \vec{u} + [I(x, y) - I(x, y-1)] \vec{v} \quad (\text{I.11})$$

Où : \vec{u} et \vec{v} deux vecteurs unitaires sur x et y.

$I(x, y)$: Fonction de niveau de gris de l'image.

Le module du gradient est :

$$|\nabla I(x, y)| = \sqrt{\left(\frac{dI}{dx}\right)^2 + \left(\frac{dI}{dy}\right)^2} \quad (\text{I.12})$$

Son orientation est :

$$\alpha = \arctg \frac{dI/dx}{dI/dy} \quad (\text{I.13})$$

7.2.2 L'approche Laplacienne

La méthode Laplacienne consiste à appliquer l'opérateur Laplacien à une image et ensuite de rechercher les zéros du laplacien. Les endroits où il change de signe signifient que l'on a un contour. De même que pour le gradient, on va ensuite effectuer un seuillage pour ne sélectionner que les passages par 0 de grande amplitude.

L'opérateur Laplacien est défini comme suit :

$$\nabla^2 I = \frac{d^2 I}{dx^2} + \frac{d^2 I}{dy^2} \quad (\text{I.14})$$

Une approximation discrète de celui-ci est donnée par la fonction $L(x,y)$ définie par :

$$L(x,y) = I(x,y) - \frac{1}{4}[I(x,y+1) + I(x,y-1) + I(x+1,y) + I(x-1,y)] \quad (\text{I.15})$$

Dans ce cas, on cherche le passage par zéro de cette fonction. La façon la plus simple est de prendre les points qui vérifient $L(x,y)=0$. Cet opérateur est représenté sous la forme matricielle suivante :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (\text{I.16})$$

7.2.3 Opérateur de Sobel

Cet opérateur combine les opérateurs de lissage et les opérateurs différentiels. Il est alors l'une des solutions du problème du bruit, il utilise les deux masques :

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (\text{I.17})$$

7.2.4 Opérateur de Prewitt

Prewitt approxime le gradient par la convolution de l'image avec les masques de convolution suivants :

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad G_y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \quad (\text{I.18})$$

7.3 Approche par classification de pixels

Dans cette approche la segmentation sera vue comme étant un problème de classification de données. Les techniques de classification identifient les classes des pixels homogènes dans l'espace des caractéristiques utilisées (niveau de gris, paramètres de texture,...) sous la forme de nuages de points opaques. Ensuite chaque classe est étiquetée donnant ainsi l'image des étiquettes.

Les méthodes de classifications ne prennent pas en compte la disposition spatiale des pixels et ne considèrent que la distribution des niveaux de gris ou couleurs dans l'espace de représentation utilisé. Elles identifient les classes de pixels en présence dans l'image et affectent à chaque pixel une étiquette indiquant la classe à laquelle il appartient. La formation des régions n'est obtenue qu'après analyse la connexité des pixels appartenant à la même classe.

Les méthodes ne prenant en compte qu'un seul attribut (niveau de gris ou une seule composante colorimétrique) sont qualifiées de méthodes monodimensionnelles, par contre, les méthodes exploitant plusieurs attributs (couleur, textures, ...) sont qualifiées de méthodes multidimensionnelles.

Il existe deux types de classifications, la classification supervisée et la classification non supervisée

La classification supervisée nécessite la présence d'un superviseur qui va apprendre au système comment classer un ensemble connu de données, puis le système sera en mesure de connaître des informations à priori sur les images à classer ou à segmenter.

La classification non supervisée a pour but de retrouver des partitions d'un ensemble de pixels en fonction d'un critère de proximité de leurs vecteurs d'attributs dans l'espace de représentation. Elles sont utilisées pour effectuer de la classification sans connaissance à priori sur l'image.

8. Définition de la texture

La définition littéraire de la texture est la "répétition spatiale d'un même motif dans différentes directions de "l'espace".

Haralick [8] élargit la définition en décrivant une texture comme un phénomène à deux dimensions: la première concernant la description d'éléments de base ou primitives (le motif) à partir desquels est formée la texture, la deuxième dimension est relative à la description de l'organisation spatiale de ces primitives.

En pratique, nous distinguons deux grandes classes de textures, qui correspondent à deux niveaux de perception: les macro-textures et les micro-textures.

Les macro-textures présentent un aspect régulier, sous forme de motifs répétitifs spatialement placés selon une règle précise (ex: peau de lézard, mur de brique) elles nécessitent une approche structurale déterministe pour leur analyse.

Les micro-textures présentent des primitives "microscopiques" distribuées de manière aléatoire (ex: sable, laine tissée, herbe).

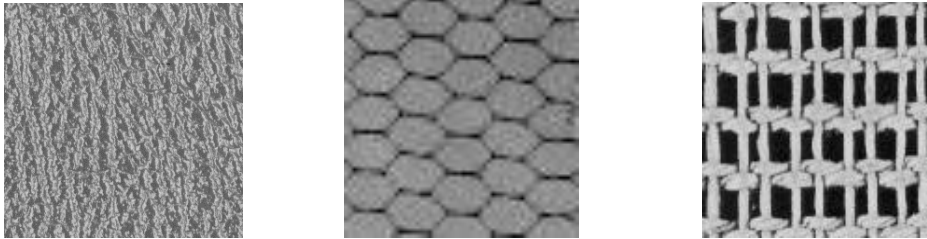


Fig.I.11: Exemple d'images texturées

9. Analyse de la texture

L'analyse de texture regroupe un ensemble de techniques mathématiques permettant de quantifier les différents niveaux de gris présents dans une image en termes d'intensité et de distribution. Le but de l'analyse de texture est de formaliser les descriptifs de la texture par des paramètres mathématiques qui serviraient à l'identifier [4].

Comme il existe deux grandes classes de texture, deux approches seront donc nécessaires: les méthodes statistiques et les méthodes structurales.

Les méthodes statistiques étudient les relations entre un pixel et ses voisins. Elles sont utilisées pour caractériser des structures fines, sans régularité apparente. L'étude se fait sur des structures tout à fait aléatoires et le plus souvent non homogènes, c'est pourquoi ce type de méthode sera préférentiellement utilisé pour les micro-textures. Dans ce cas, la texture est décrite par les statistiques de la distribution de ces niveaux de gris (ou intensité).

Les méthodes structurales tiennent compte de l'information structurale et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques. Elles permettent de décrire la texture en définissant les primitives et les

"règles" d'arrangement qui les relient. En effet, les textures ordonnées possèdent des primitives qui se répètent dans les images en des positions selon une certaine loi.

10. Conclusion

A travers ce chapitre nous avons essayé de faire une introduction générale au traitement d'images, en définissant les différents types d'images et les caractéristiques qui s'y rattachent. Nous avons aussi présenté quelques techniques de segmentations parmi les plus répondues. Puis, dans le chapitre qui va suivre, nous étudierons en détail et testerons un algorithme particulier de segmentation qui est JSEG.

Chapitre II

Segmentation d'images texturées par JSEG

1. Introduction

La méthode de segmentation des images couleurs texturées nommée JSEG est une méthode récente développée par les chercheurs Y. Deng et B. S.Manjunath en 2001.

Le processus de segmentation est séparé en deux étapes qui sont, la quantification de l'espace des attributs couleurs et la segmentation spatiale.

Cette méthode, initialement développée pour des images couleurs s'adapte facilement à des images en niveau de gris, il suffira pour cela, durant l'étape de quantification, de considérer pour chaque pixel la valeur de son niveau de gris au lieu de prendre un vecteur de trois valeurs.

Dans ce chapitre, nous expliquerons la méthode de segmentation JSEG en détaillant chaque technique utilisée dans les différentes étapes et en fournissant à chaque fois des exemples d'illustration.

2. Principe de la segmentation JSEG

L'algorithme JSEG présente deux étapes principales et indépendantes qui sont :

- La quantification couleur (ou niveau de gris).
- La segmentation spatiale.

Dans la première étape, les différentes couleurs (ou niveaux de gris) de l'image sont quantifiées en un nombre de classes représentatives de cet espace couleur (ou niveau de gris).

Cette image des étiquettes, nommée class-map, sera ensuite considérée comme une représentation des textures composant l'image.

Après cette quantification des couleurs (ou des niveaux gris), une segmentation spatiale sera réalisée sur le class-map en ne faisant plus intervenir l'information couleur(ou niveau de gris). La figure Fig.II.1 fournit une représentation schématique de l'algorithme JSEG.

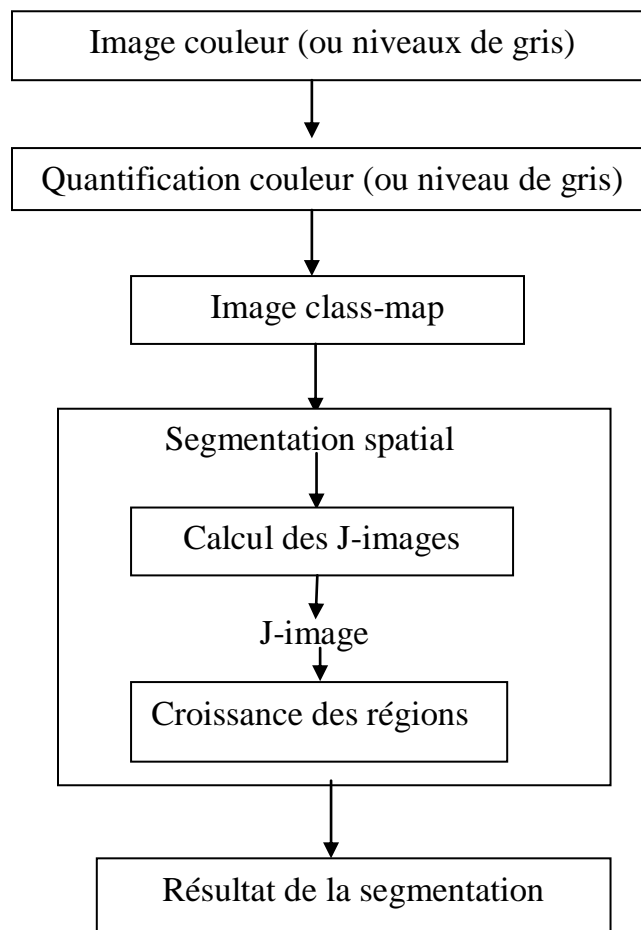


Fig.II.1: Schéma de l'algorithme de segmentation d'image couleurs (ou niveau de gris) JSEG [4]

3. Quantification couleur (ou niveau de gris)

La quantification couleur (ou niveau de gris) a pour but de réduire le nombre de couleurs (niveau de gris) présentes dans l'image. Cette réduction ne se fera pas de manière uniforme car l'œil humain est plus sensible aux variations contenues dans des régions homogènes qu'aux variations présentes dans des régions texturées. Ainsi les couleurs ou les niveaux de gris seront quantifiées de manière grossière dans les régions texturées sans affecter de manière significative la perception visuelle.

Pour exploiter cette notion, on affecte à chaque pixel de l'image, un poids tel que celui-ci sera plus important pour les pixels se trouvant dans des régions homogènes et moins important pour les pixels se trouvant dans des régions fortement texturées.

L'algorithme « Peer Group Filtring » ou P.G.F. est utilisé pour le calcul de ces poids perceptuels. Ainsi pour chaque pixel n de l'image, ayant pour valeur $x(n)$, on calcul son poids perceptuel $v(n)$ (le calcul est détaillé dans le paragraphe 3.1).

Ensuite, en pondérant les pixels par leur poids perceptuel, on les classe en utilisant l'algorithme de «Lloyd généralisé » ou «generalized Lloyd algorithm » (GLA). Cet algorithme consiste à diviser, à chaque itération, la classe présentant une distorsion maximale, en partant d'une classe unique formée de tous les pixels de l'image comme étape initiale et en s'arrêtant lorsqu'un critère d'arrêt sera satisfait. Au final, on obtient l'image quantifiée appelée « class-map ».

En résumé, la quantification couleur est réalisée en deux étapes qui sont :

- Le calcul des poids perceptuels.
- La quantification perceptuelle.

3.1 Calcul des poids perceptuels

Afin de calculer les poids perceptuels $v(n)$, l'algorithme PGF se base sur le calcul des deux valeurs principales $T(n)$ et $P(n)$ qui représentent respectivement la valeur maximale des distances calculées pour une région locale et le groupe de pixels semblables. Le calcul des valeurs $T(n)$ et $P(n)$ se fait en utilisant une fenêtre de dimension $w*w$ centrée sur tous les pixels de l'image.

Dans toutes les étapes qui vont suivre $x_0(n)$ est considéré comme étant un vecteur de trois couleurs pour des images couleurs ou simplement la valeur du niveau de gris du pixel considéré.

Le calcul des poids perceptuels $v(n)$ doit suivre les étapes suivantes :

1. Centrer le pixel $x_0(n)$ dans une fenêtre de dimension $w * w$ (notons que w doit être un nombre impair).

2. Mesurer les distances euclidiennes $d_i(n)$ entre les pixels $x_i(n)$ de la fenêtre et le pixel central $x_0(n)$, tel que $i=0 \dots, k=w^2-1$.

$$d_i(n) = \|x_0(n) - x_i(n)\| \quad i=0 \dots k \quad (\text{II.1})$$

3. Classer les distances d_i dans un ordre croissant.

$$d_0(n) \leq d_1(n) \leq \dots \leq d_k(n) \quad (\text{II.2})$$

4. Rechercher l'ensemble des pixels $P(n)$ de tailles $m(n)$, semblables à $x_0(n)$.

(Cette étape est détaillée plus loin).

5. Calculer la distance maximale appelée aussi distance seuil notée « $T(n)$ » pour chaque groupe de pixels $P(n)$ tel que :

$$T(n) = d_{m(n)-1}(n) \quad (\text{II.3})$$

6. Calculer le poids perceptuel $V(n)$ de chaque pixel de l'image en utilisant la relation suivante :

$$V(n) = \exp(-T(n)) \quad (\text{II.4})$$

Pour ce qui est de l'étape 4, on recherche l'ensemble des pixels $P(n)$. Cet ensemble est constitué du pixel central $x_0(n)$ et des pixels voisins dont les niveaux de gris lui sont similaires.

L'ensemble $P(n)$ est de taille $m(n)$, il est défini par :

$$P(n) = \{x_i(n), i = 0, \dots, m(n) - 1\} \quad (\text{II.5})$$

La taille $m(n)$ doit respecter la condition suivante :

$$d_{m(n)-1}(n) \leq T(n) \text{ et } d_{m(n)}(n) > T(n) \quad (\text{II.6})$$

Comme il est impossible de calculer une valeur fixe de $T(n)$ qui soit optimale pour toute l'image on utilise le discriminant de Fisher pour calculer le seuil $T(n)$.

Pour cela on maximise le critère J suivant :

$$J(i) = \frac{|a_1(i) - a_2(i)|^2}{s_1^2(i) + s_2^2(i)} \quad i=1 \dots k \quad (\text{II.7})$$

$$\text{Où :} \quad a_1(i) = \frac{1}{i} \sum_{j=0}^{i-1} d_j(n) \quad (\text{II.8})$$

$$a_2(i) = \frac{1}{k+1-i} \sum_{j=i}^k d_j(n) \quad (\text{II.9})$$

$$s_1^2(i) = \sum_{j=0}^{i-1} |d_j(n) - a_1(i)|^2 \quad (\text{II.10})$$

$$s_2^2(i) = \sum_{j=i}^k |d_j(n) - a_2(i)|^2 \quad (\text{II.11})$$

La valeur de i qui maximise le critère J correspond à la valeur $m(n)$ donnant la taille de $P(n)$. Ainsi :

$$m(n) = \arg(\max_i J(i)) \quad (\text{II.12})$$

Afin d'illustrer les différents calculs réalisés dans ce paragraphe, nous fournissons deux exemples d'images en niveau de gris ainsi que la représentation des valeurs $T(n)$ et $V(n)$ de chaque pixel sous forme d'images en figures II.2 et II.3.

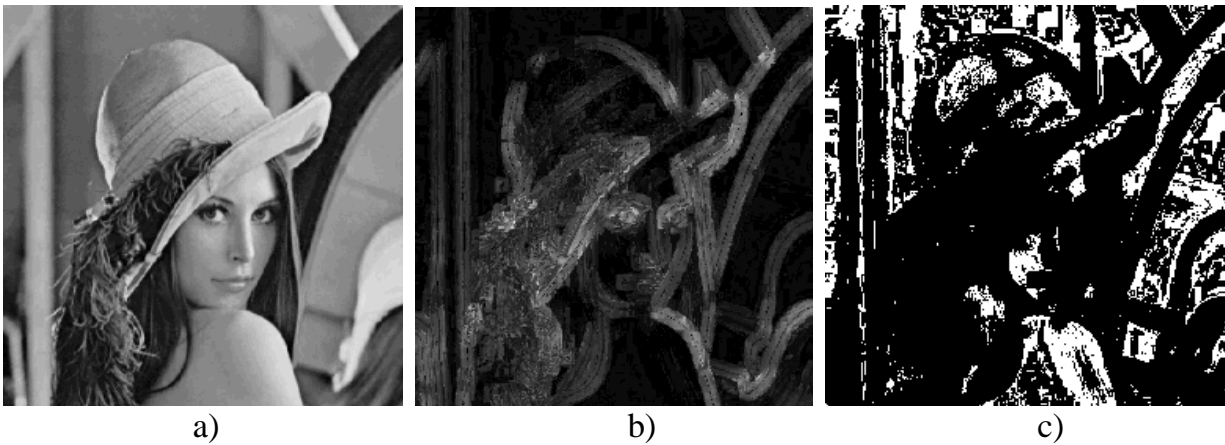


Fig. II.2 : Images Lenna, $T(n)$ et $V(n)$

- a) image original Lenna
- b) Représentation sous forme d'image des distances maximales $T(n)$
- c) Représentation sous forme d'image de poids perceptuels $V(n)$

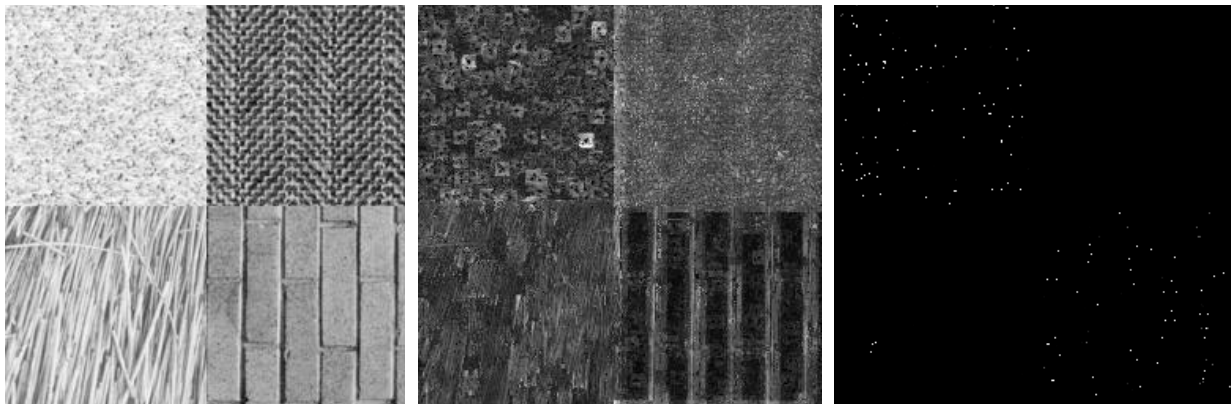


Fig. II.3 : Images synthétique 1, $T(n)$ et $V(n)$

- a) image original synthétique 1
- b) Représentation sous forme d'image des distances maximales $T(n)$
- c) Représentation sous forme d'image de poids perceptuels $V(n)$

Il faut noter que les valeurs de $V(n)$ sont faibles dans les régions texturées ce qui explique les valeurs importantes de $T(n)$ en ces régions.

3.2 La quantification perceptuelle

Après le calcul des poids perceptuels de tous les pixels de l'image, on réalise une quantification des couleurs (ou niveaux de gris) en pondérant chaque pixel par son poids perceptuel.

Les étapes de cette quantification sont :

1. Calculer la moyenne T_{avg} des distances maximales $T(n)$ obtenues après l'application de l'algorithme PGF de l'étape précédente.

2. Estimer le nombre de classes initiales N dans la quantification par la relation suivante :

$$N = \beta T_{avg} \quad (\text{II.13})$$

Où β est égal à 2 (valeur expérimentale).

3. Classer les pixels de l'image par l'algorithme de Lloyd généralisé « generalized Lloyd algorithm » ou (GLA) en prenant en considération les poids perceptuels des différents pixels de l'image. Les détails de cette étape sont donnés plus bas. A la fin de cette étape, on obtient N centres de classes et chaque pixel est affecté au centre le plus proche.

4. Recalculer les vraies centres de classes c'est-à-dire sans appliquer les poids $V(n)$ sur les pixels.

5. Affecter les pixels de l'image du point de vue de leur niveau de gris aux différentes classes en utilisant la distance euclidienne.

6. Fusionner les classes dont les centres de classes ont une distance inférieure à un seuil pré-régulé QT , ce seuil est appelé seuil de quantification.

7. Etiqueter les pixels de l'image selon la classe dans laquelle sont contenus leurs niveaux de gris. Ce qui permettra d'obtenir une carte des classes de l'image ou le class-map.

L'application de l'algorithme GLA pour la classification des pixels de l'image (étape 3 précédente) se fait de la manière suivante :

1. Déterminer un centre de classe initial calculé pour toute l'image de la manière

$$c = \frac{\sum v(n)x(n)}{\sum v(n)}; \quad x(n) \in I \quad (\text{II.14})$$

Considérer l'image comme étant la classe présentant une distorsion maximale.

2. Diviser la classe présentant une distorsion maximale en deux classes.

3. Calculer les centres c_i et les distorsions D_i des différentes classes C_i présentes dans l'image en utilisant les formules suivantes :

$$c_i = \frac{\sum v(n)x(n)}{\sum v(n)}, \quad x(n) \in C_i \quad (\text{II.15})$$

$$D_i = \sum_n v(n) \|x(n) - c_i\|^2, \quad X(n) \in C_i \quad (\text{II.16})$$

4. Si le nombre de classes atteint est N finir, sinon aller à 2.

Pour illustrer cette étape de quantification, nous fournissons deux exemples sur les figures II.4 et II.5 avec différentes valeurs du seuil de quantification QT et en fournissant à chaque fois le nombre de classes détecté Nc.



Image originale



QT=2 Nc=44



QT=10 Nc=14

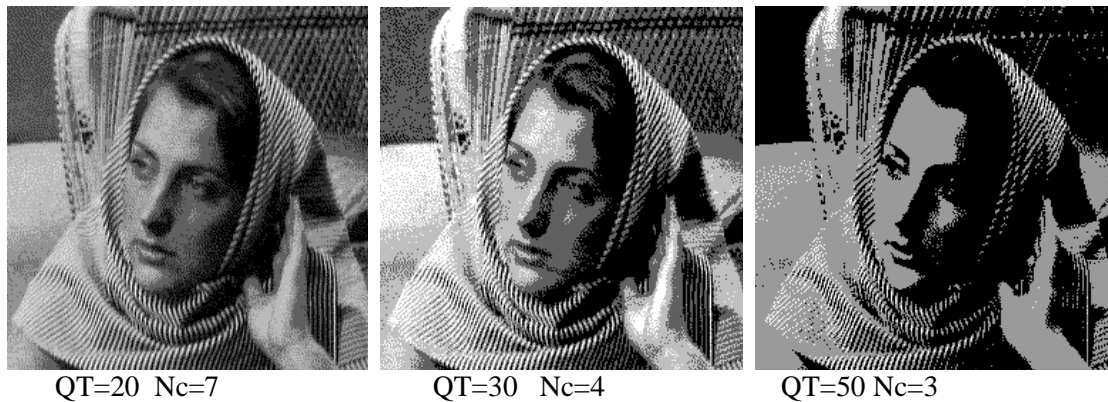


Fig. II.4 : Quantification de l'image Woman avec différents seuil de quantification

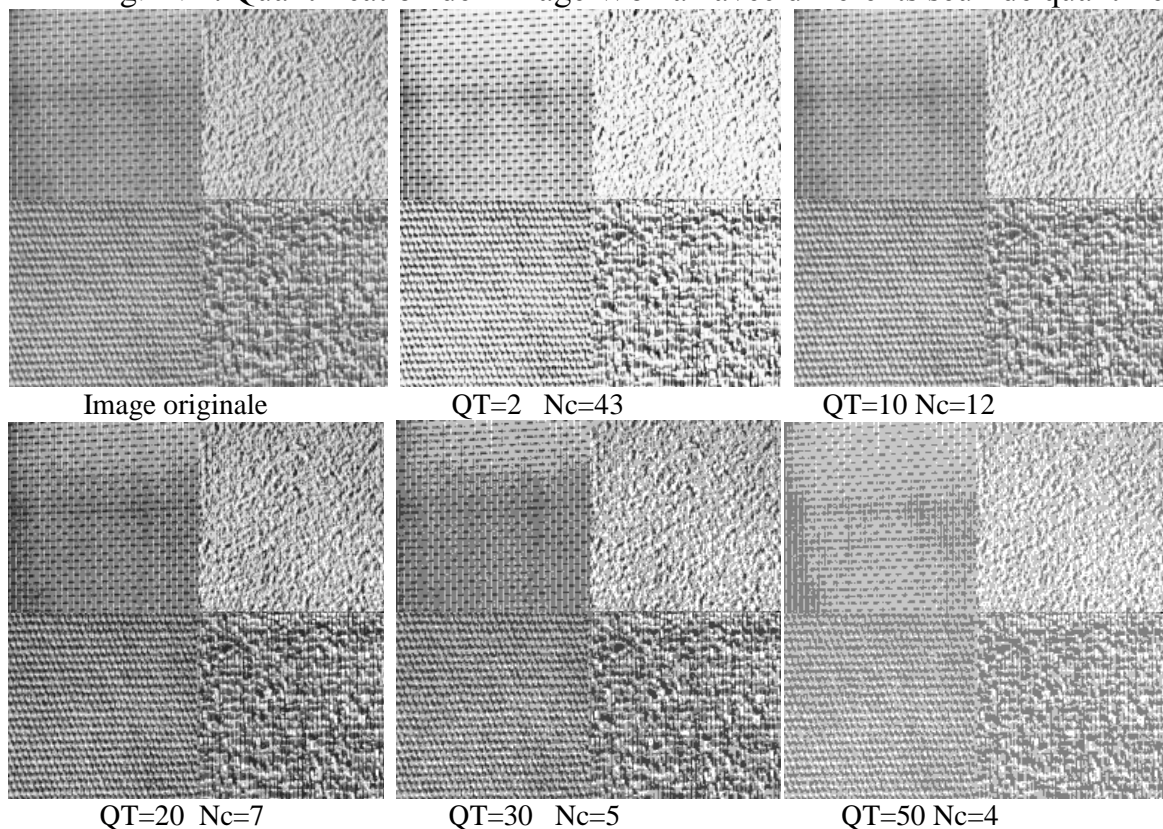


Fig. II.5 : Quantification de l'image synthétique 2 avec différents seuil de quantification

4. Segmentation spatial

Après avoir obtenu la carte, considérée class-map de l'image à la fin de l'étape de quantification, l'étape de segmentation spatiale travaillera seulement avec cette carte des classes comme étant un ensemble de points de données spatiales localisées dans un espace 2D de coordonnées (x,y).

La segmentation spatiale est réalisée sur les J-images à différentes échelles qui correspondent à des images où chaque pixel représente une mesure « J » d'homogénéité. Nous allons donc dans ce paragraphe, commencer par donner la méthode utilisée pour le calcul de la mesure J, ensuite expliquer comment calculer les J-images à différentes échelles puis développer l'algorithme de segmentation.

4.1 La mesure J

Au cours de cette partie, nous avons à calculer un critère J qui présente le rapport entre les distances interclasses S_B et les distances intra-classes S_W .

Supposons que l'on souhaite calculer la valeur J d'une partie du class-map. Cette carte des classes peut être de même dimension ou de dimension plus faible que le class-map.

Soit Z l'ensemble de tous les points N de cette carte de classes. Soit $z=(x, y)$ la position spatiale d'un point de cette carte tel que $z \in Z$.

Supposons que Z est subdivisé en C classes Z_i , avec $i=1 \dots \dots C$.

Le critère J calculé pour cette carte des classes est donné par la relation:

$$J = \frac{S_B}{S_W} = \frac{S_T - S_W}{S_W} \quad (\text{II.17})$$

Avec S_B : variance interclasses.

S_W : variance intra-classes.

S_T : variance globale.

Les différentes variances sont données par les relations :

$$S_T = \sum_{z \in Z} \|z - m\|^2 \quad (\text{II.18})$$

$$S_W = \sum_{i=1}^C \sum_{z \in Z_i} \|z - m_i\|^2 \quad (\text{II.19})$$

Où m représente la moyenne de tous les éléments z de la carte des classes et m_i est la moyenne des éléments de Z_i c'est-à-dire des éléments appartenant à la classe i .

m et m_i sont décrits par les formules :

$$m = \frac{1}{N} \sum_{z \in Z} z \quad (\text{II.20})$$

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} z \quad (\text{II.21})$$

Pour que la notion de la mesure J soit claire, nous nous sommes donné trois cas de cartes des classes différentes dont la segmentation est très facile à opérer manuellement.

Ces trois cartes des classes contiennent chacune 3 classes. On a assigné à chaque pixel les valeurs 1, 2 ou 3 correspondant à la classe du pixel.

On a aussi indiqué les valeurs de J pour chaque carte de classes sur la figure II.6.

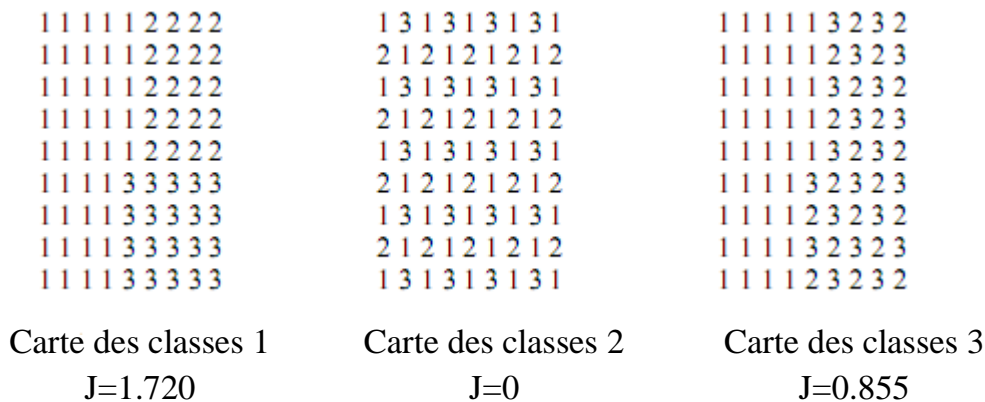


Fig.II.6: Exemple de carte de classes

On voit clairement en figure II.6 que pour la carte des classes 1, il y a une seule segmentation possible qui consiste en 3 régions chacune contenant des pixels appartenant à une seule classe. La carte des classes 2 est déjà uniforme et aucune segmentation n'est possible. Pour la carte des classes 3, une bonne segmentation consiste en 2 régions, l'une contenant la classe 1 et l'autre contenant les classes 2 et 3 (voir figure II.7). Il faut noter que pour ces 3 différents cas, nous obtenons des valeurs très différentes pour J .

Dans l'étape suivante, on va recalculer J pour chaque région au lieu de le calculer sur la carte entière, on introduit également la moyenne \bar{J} , définie comme suit :

$$\bar{J} = \frac{1}{N} \sum_k M_k J_k \tag{II.22}$$

Où J_k est la valeur de J pour la région k, M_k est le nombre de points de la région k et N est le nombre de points de la carte des classes.

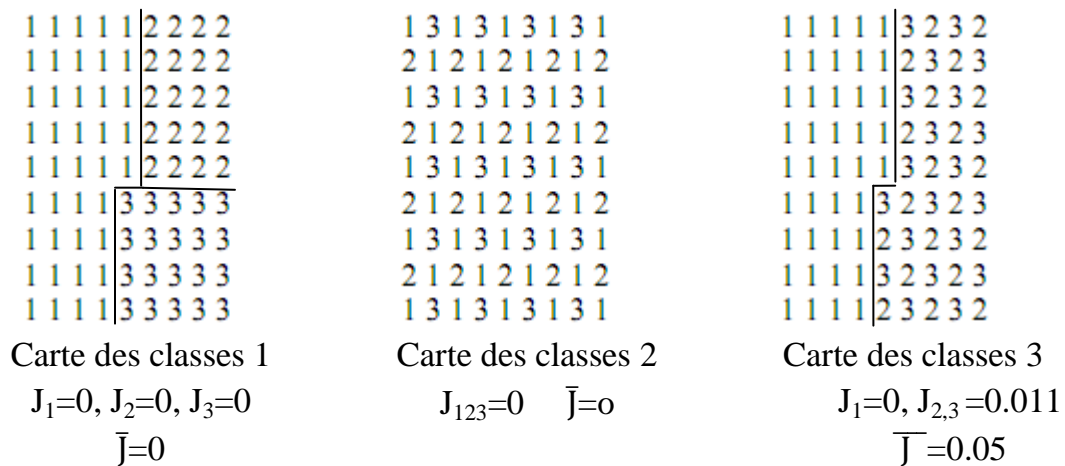


Fig.II.7: Résultat de la segmentation manuelle des 3 cartes de classes de la figure II.6

Pour la carte des classes 1 $\bar{J}=0$, c'est la valeur minimale que peut prendre \bar{J} donc la carte est bien segmentée car les trois régions résultantes sont homogènes. La carte 3 admet une valeur $\bar{J}=0.05$ qui n'est pas nulle car la deuxième région n'est pas uniforme (contient deux classes 2 et 3).

\bar{J} est donc le critère à minimiser pour obtenir une bonne segmentation. En effet, pour un nombre fixé de régions, la meilleure segmentation est celle qui a la plus basse valeur de \bar{J} . Dans le cas d'une «bonne» segmentation, chaque région est uniforme et contient un petit nombre de classes, la valeur de J pour chaque région est petite et donc la valeur de \bar{J} est également petite. Cependant, le critère \bar{J} n'est pas très facile à manipuler dans la mesure où il faut le minimiser pour l'image entière. Il est alors plus simple de manipuler le critère J directement car, appliqué localement à une petite zone de l'image, il permet d'indiquer si cette zone est près d'une frontière ou plutôt au centre d'une région.

4.2 Détermination des J-images

Une J-image est une image dont la valeur des pixels est donnée par la valeur de J calculée pour une petite zone ou fenêtre centrée sur ce pixel.

Les dimensions des fenêtres locales utilisées, déterminent les échelles et les dimensions des régions de l'image qui peuvent être détectées.

La plus petite échelle est notée échelle 1 et est de dimension 9*9. La plus grande est notée échelle 4 et est de dimension 65*65. La figure II.8 illustre les différentes échelles ainsi que leurs dimensions.

Echelle	Fenêtre (pixels)	Échantillonnage (1/pixels)	Dimension de la région (pixels)	Min. germe (Pixels)
1	9x9	1/ (1x1)	64x64	32
2	17x17	1/ (2x2)	128x128	128
3	33x33	1/ (4x4)	256x256	512
4	65x65	1/ (8x8)	512x512	2048

Fig.II.8: Dimensions de la fenêtre à différentes échelles.

Les fenêtres de petites dimensions sont utiles dans la localisation des contours intensité, tandis que les grandes fenêtres sont utiles pour détecter les frontières des textures.

Les coins des fenêtres sont supprimés pour les rendre circulaire de façon à ce que la fenêtre n'ait aucun effet sur les objets rectangulaires.

Quelque soit la taille de la fenêtre, on doit se ramener à une fenêtre de base de taille 9*9 sans les coins avec un taux de prélèvement qui est spécifié sur la troisième colonne du tableau de la figure II.8.

La figure en dessous montre une fenêtre à l'échelle 2, où le taux de prélèvement est de 1 pixel sur 2 suivant les directions X et Y. Seuls les points '+' sont utilisés pour le calcul des valeurs J locales.

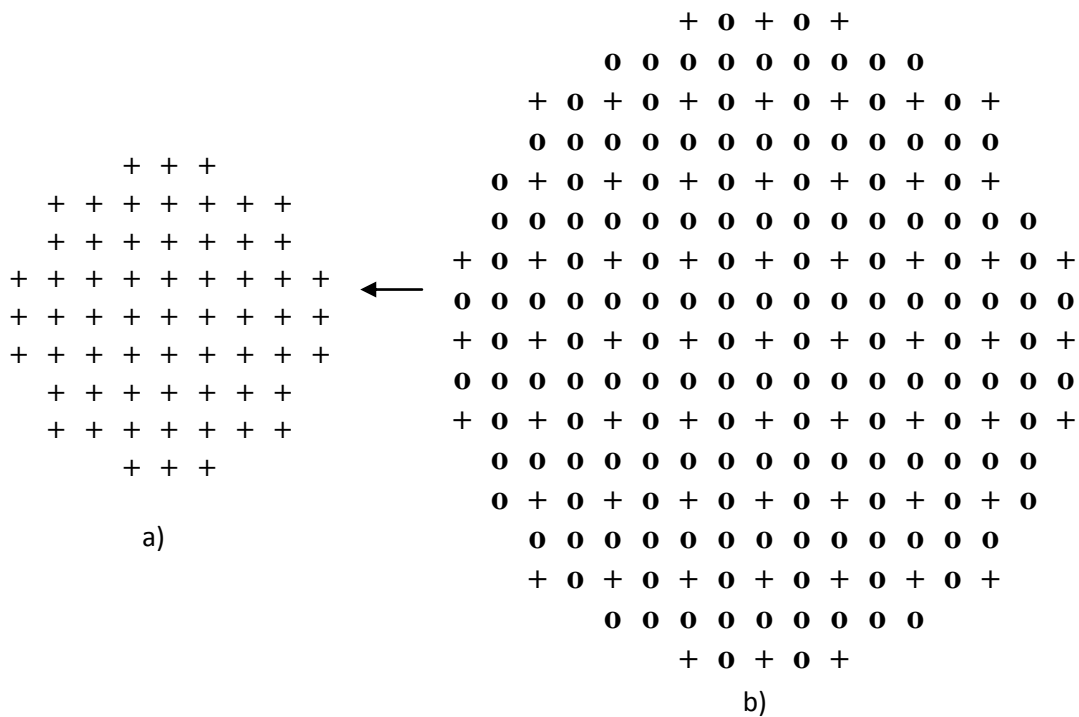


Fig.II.9: Extraction de la fenêtre de base à partir d'une fenêtre à l'échelle 2
 a) Fenêtre de base 9*9 b) Fenêtre à l'échelle 2

Sur le tableau de la figure II.8 est représenté en colonne 4 les dimensions des régions souhaitables pour chaque échelle. Par exemple, si les dimensions de l'image sont supérieures à 256*256 tout en restant inférieures à 512*512, l'échelle de départ est 3.

Le nombre d'échelles nécessaires à la segmentation est fourni par l'utilisateur. Ceci détermine l'échelle minimale qui termine le programme.

Dans le cas de l'exemple précédent, c'est-à-dire dans le cas où la dimension de l'image est comprise entre 256×256 et 512×512 et si le nombre d'échelles est fixé à 2, on déterminera alors les J-images à l'échelle 3 et à l'échelle 2.

Sur les figures ci-dessous, des exemples de J-images à différentes échelles sont illustrées.

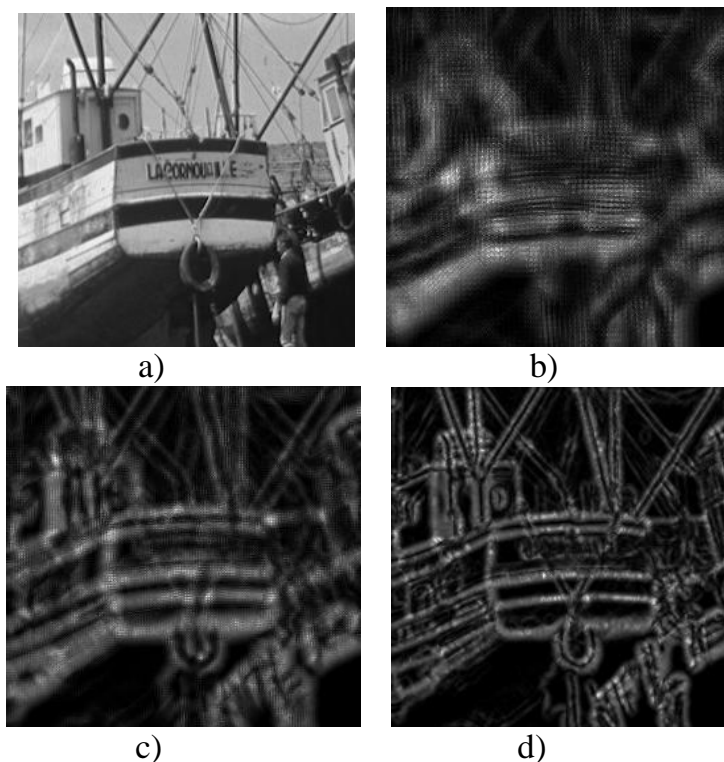
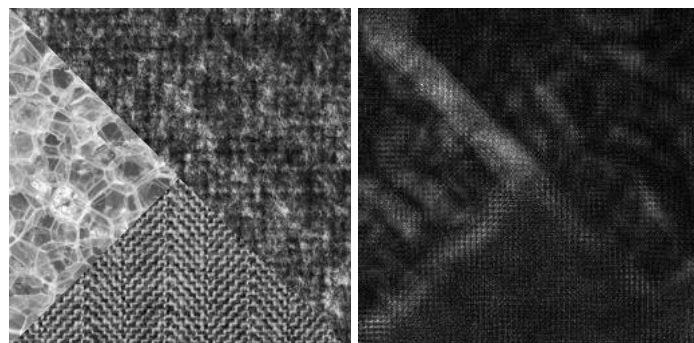


Fig. II.10 : Image Bateau et ses J-images à différentes échelles
a) Image originale b) J-image à l'échelle 3 c) J-image à l'échelle 2 d) J-image à l'échelle 1



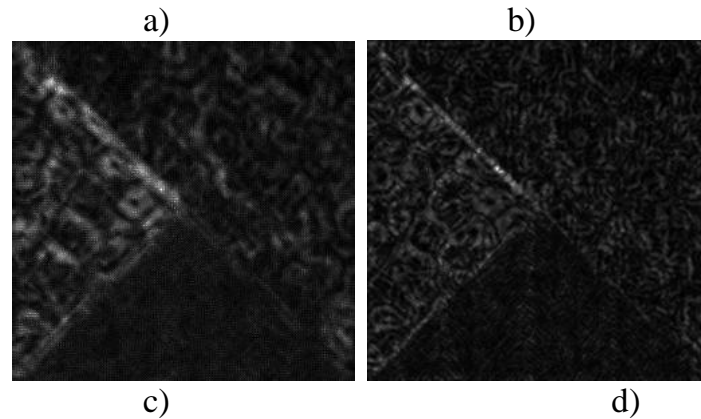


Fig. II.11 : Image synthétique 3 et ses J-images à différentes échelles

a) Image originale b) J-image à l'échelle 3 c) J-image à l'échelle 2 d) J-image à l'échelle 1

4.3 Algorithme de segmentation

Après la détermination des J-images, on procède à la segmentation spatiale en utilisant ces J-images.

On considère en premier lieu la J-image à l'échelle maximale, on réalise une segmentation en trois étapes qui consiste en :

- détermination des germes,
- croissance de germes et
- fusion des régions similaires.

On répète ensuite le même processus, c'est-à-dire les trois étapes de segmentation sur les régions obtenues à l'échelle précédente mais en prenant en considération les valeurs de J à l'échelle actuelle. Toutes les régions de l'échelle précédente doivent être traitées. Ce processus se répète jusqu'à atteindre l'échelle minimale.

Nous allons donc développer les trois étapes de l'algorithme de segmentation spatiale.

4.3.1 Détermination des germes

Rappelons qu'une valeur élevée de J indique que le pixel est près d'une frontière et inversement une basse valeur de J indique que le pixel est plutôt au centre d'une région.

Comme la segmentation à réaliser est de type croissance de régions, il faut initialement sélectionner les germes de ces régions.

Les petites régions dites régions germes sont alors les régions où les valeurs locales de J sont minimales. Un ensemble de germes est déterminé au début pour servir de base à la segmentation en utilisant l'accroissement des régions.

Pour trouver le meilleur ensemble de germes dans une région on procède de la manière suivante:

1. Calculer la moyenne μ_j et de la variance σ_j de J à l'échelle considérée.

2. Déduire le seuil T_j pour chaque région, $T_j = \mu_j + a \sigma_j$.

a est choisi parmi l'ensemble de valeurs $\{-0.6, -0.4, -0.2, 0, 0.2, 0.4\}$ de manière à ce que le nombre de germes obtenus soit maximal.

Les pixels dont J est inférieur à T_j sont considérés comme des points germes.

La figure ci-dessous illustre un exemple de régions germe déterminées sur l'image Cameraman. Les régions germes sont représentées en blancs alors que les autres pixels sont représentés en noir.

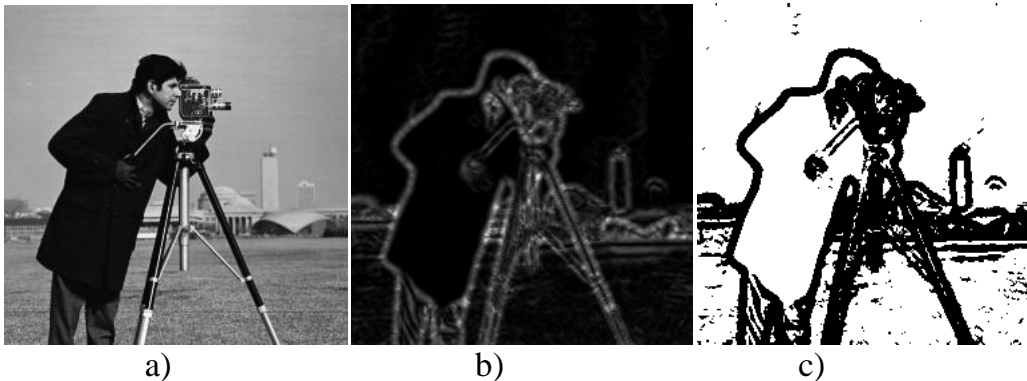


Fig. II.12 : Image Cameraman sa J image et la représentation des régions germe
a) Image originale b) J -image à l'échelle 3 c) image des germes

4.3.2 Croissance des germes

Les régions seront obtenues en faisant croître les germes déterminés dans l'étape précédente en suivant les étapes ci-dessous.

1. Enlever les « trous » présents dans les germes.
2. Déterminer la moyenne des J des pixels non affectés à des régions.
3. Affecter les pixels présentant des valeurs de J inférieures à cette moyenne à la région croissante qui lui est adjacente si celle-ci est unique.
4. Répéter l'étape 2 jusqu'à ce que tous les pixels de l'image soient classés.

La figure II.13 fournit une représentation des régions de l'image Caméraman après cette étape de croissance des germes. L'image des germes est aussi donnée en fausse couleurs pour mieux visualiser les régions germes. Les pixels représentés en bleu foncé sont les pixels non classés.

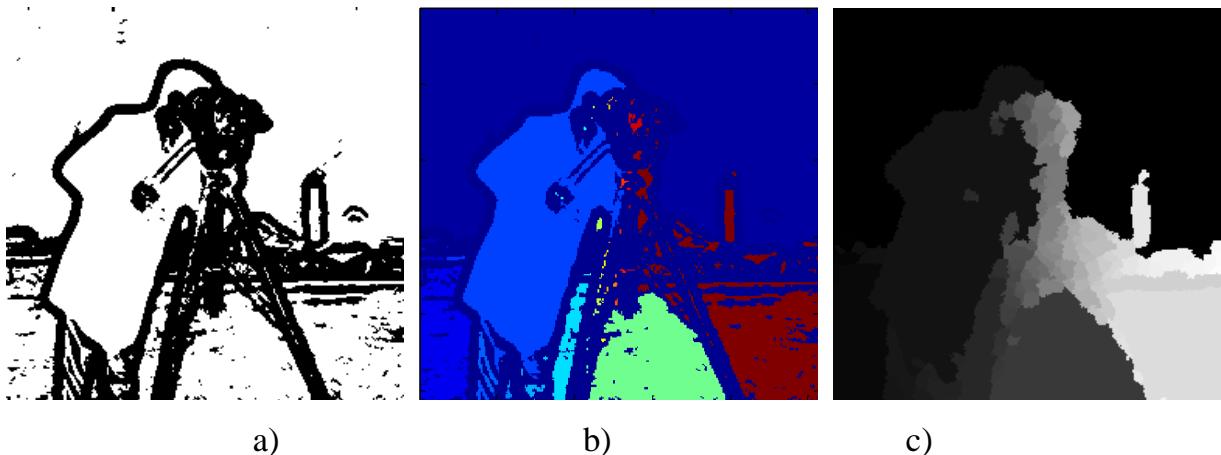


Fig. II.13 : Représentation de a) image des germes b) image des régions germe et de c) l'image des régions de l'image Cameraman

4.3.3 Fusion des régions

Après l'étape de croissance des germes, on obtient une image segmentée où le nombre de régions est souvent important. Ces régions seront fusionnées en tenant compte de leur similarité couleur ou niveau de gris. Les étapes de ce processus sont :

1. Déterminer la valeur en niveau de gris ou en couleur pour chaque région.
2. Calculer la distance entre les régions voisines.

3. Les paires de régions présentant une distance minimale et inférieure à un seuil S sont fusionnées.
4. Calculer les nouvelles valeurs en niveau de gris ou en couleur pour les nouvelles régions.
5. Répéter à partir de 2 jusqu'à ce que les distances entre les régions voisines soient toutes supérieures à S .

Sur la figure ci-dessous, nous donnons les résultats obtenus après la fusion des régions avec différentes valeurs du seuil S .

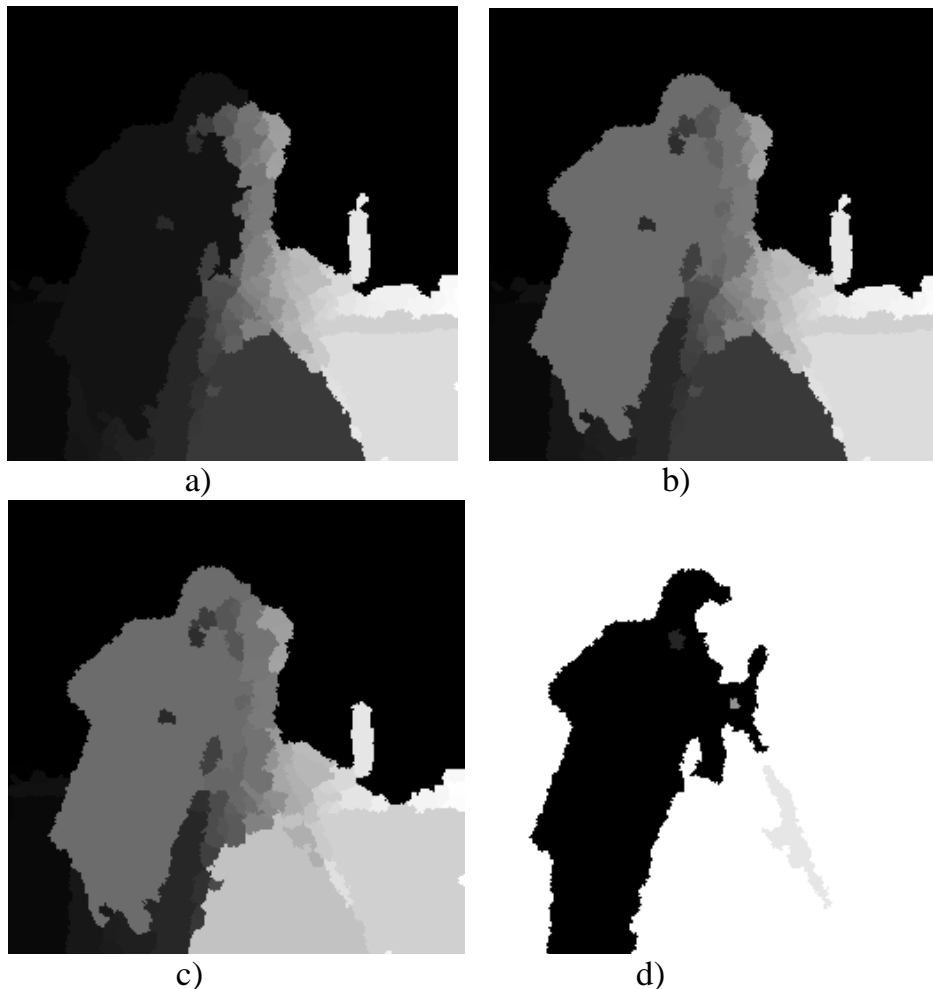


Fig. II.14 : Résultats obtenu après regroupement des régions dans l'image Cameraman
a) image des régions b) regroupement avec $S=2$ c) regroupement avec $S=10$ d) regroupement avec $S=100$

5. Conclusion

Dans ce chapitre, nous avons développé la méthode de segmentation JSEG qui est souvent utilisée pour segmenter des images texturées.

Cette méthode est en fait une succession de techniques qui, utilisée dans l'ordre présenté, permet de tirer avantage de chacune d'elles.

Nous avons pu programmer chacune de ces étapes, dans le chapitre suivant, nous les appliquerons pour à la segmentation d'images réduites par les ondelettes.

Chapitre III

Théorie des ondelettes

1. Introduction

Traiter un signal donné consiste à le transformer en une forme plus appropriée pour l'utilisation envisagée en se basant sur les outils mathématiques. L'un de ces outils le plus connu est la transformée de Fourier qui permet d'explorer la composition fréquentielle d'un signal.

La notion de localisation temporelle disparaît dans l'espace de Fourier d'où la nécessité de trouver des transformations qui renseignent sur le contenu fréquentiel tout en préservant la localisation temporelle. La transformée en ondelettes est l'une de ces transformations qui s'est largement développée et qui a été appliquée pour résoudre de nombreux problèmes de traitement de signal.

Dans ce chapitre, nous rappellerons d'abord les différentes notions relatives à la transformée de Fourier, nous développerons ensuite la transformée en ondelettes en donnant des exemples d'illustrations.

2. La transformée de Fourier

Au 19^{ième} siècle (1822), le mathématicien français J. Fourier, a prouvé que n'importe quelle fonction périodique peut être exprimée comme une somme infinie de fonctions exponentielles complexes périodiques.

J. Fourier a donc introduit l'analyse fréquentielle afin d'accéder à toutes les informations contenues dans un signal et ainsi faire apparaître plus clairement ses caractéristiques entre autres fréquentiels.

La T.F (transformée de Fourier) d'un signal $f(t)$ est notée $F(\omega)$ et est donnée par :

$$T.F (f(t))=F(\omega)=\int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt \quad (\text{III.1})$$

$F(\omega)$ transformée de Fourier de $f(t)$

$\omega=2\pi\nu$ ω et ν sont respectivement la pulsation et la fréquence du signal $f(t)$.

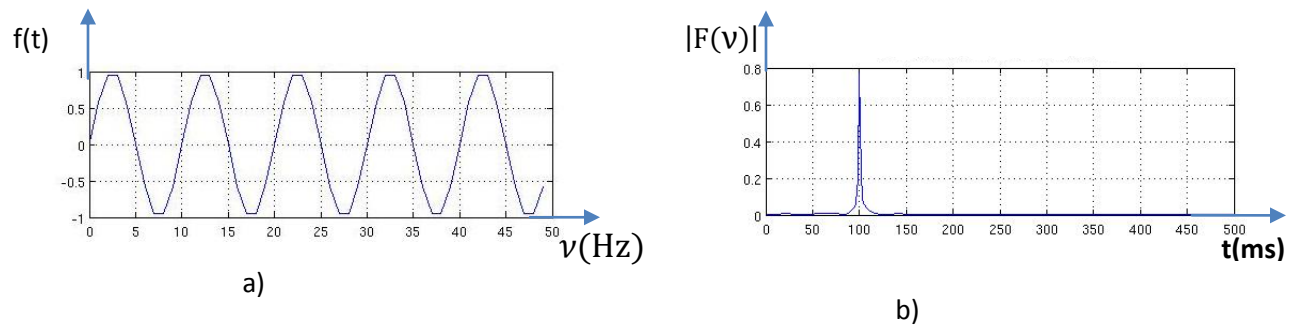


Fig.III.1: Exemple de signal périodique et de sa TF

- a) Exemple de signal périodique.
- b) Représentation de la T.F du signal en a).

L'intégrale s'effectue sur toutes les valeurs du temps (-inf. à +inf.) peu importe où la composante (fréquence) apparaît, le résultat n'en sera pas affecté.

Les fréquences mesurées ne peuvent être situées dans le temps.

La correspondance temps-fréquence inexistante c'est-à-dire deux signaux différents peuvent avoir la même transformée de Fourier (Fig.III.2).

La transformée de Fourier n'est efficace que pour des signaux stationnaires qui ne sont que des signaux dont le contenu en fréquence ne change pas au cours du temps c'est-à-dire qu'il ne dépend pas du temps.

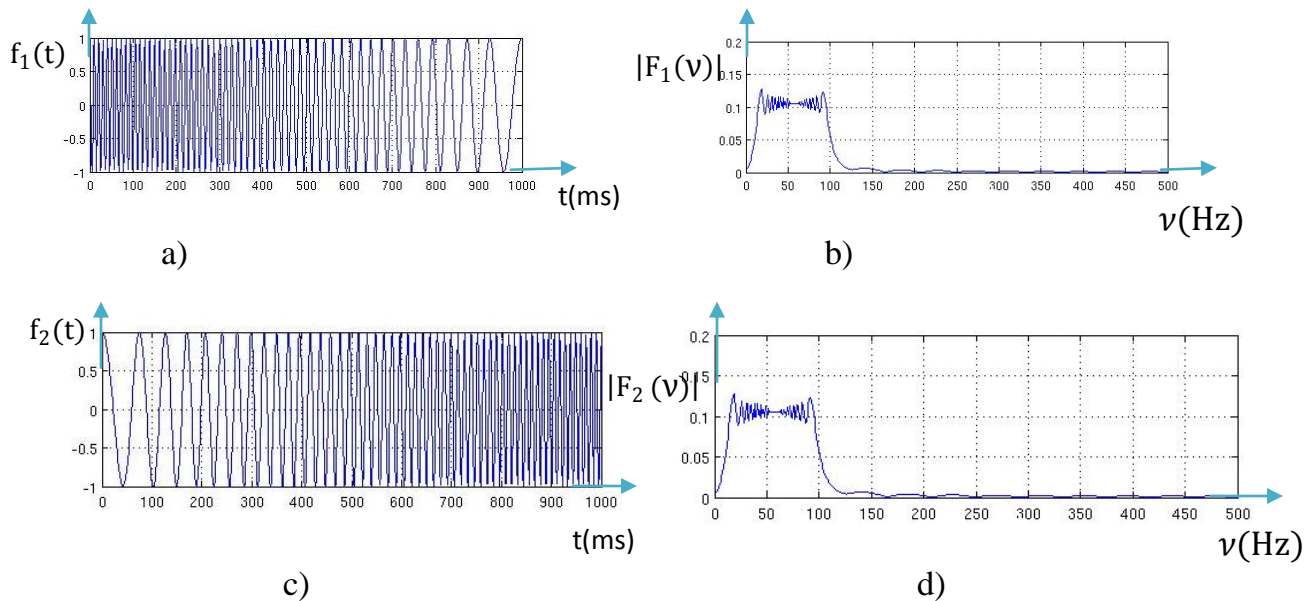


Fig.III.2 : Exemple de signaux différents possédant la même transformée de Fourier.

- a) Exemple de signal $f_1(t)$ périodique à fréquence variable.
- b) La transformée de Fourier du signal représenté en a).
- c) Exemple d'un signal $f_2(t)$ périodique à fréquence variable.
- d) La transformée de Fourier du signal représenté en c).

Pour palier à ces différents inconvénients on introduit la transformée de Fourier à fenêtre glissante dont le principe est de faire glisser une fenêtre le long du signal et d'appliquer la TF à l'intérieur de la fenêtre (voir Fig. III.3).

La TF à fenêtre glissante s'écrit sous la forme suivante :

$$\text{TFglis}(t, \omega) = \int_{-\infty}^{+\infty} f(\tau)g(\tau - t)e^{-j\omega\tau} d\tau \quad (\text{III.2})$$

Où g est la fonction fenêtre et $g(\tau - t)$ est centrée sur t .

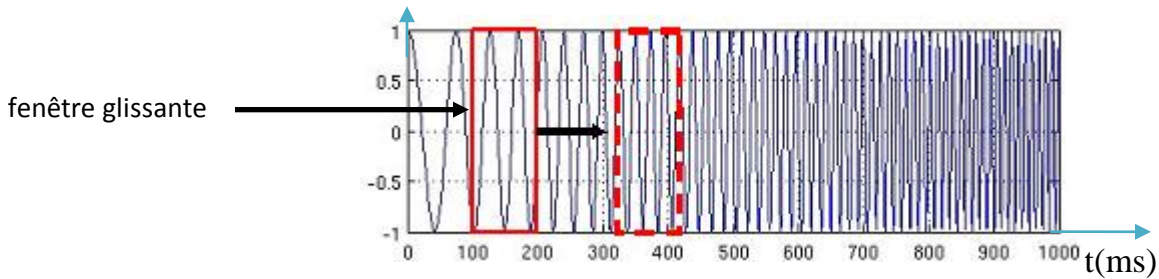


Fig.III.3: Exemple de fenêtre glissante

Plusieurs types de fenêtres peuvent être utilisés, l'une des plus connues est la fenêtre de Gabor dont la réponse en fréquence est représentée en Fig.III.4. Sa formule est la suivante :

$$g(t) = \frac{1}{2\sqrt{\pi\alpha}} \exp\left(\frac{t^2}{4\alpha}\right) \quad (\text{III.3})$$

La transformée de Fourier en utilisant la fenêtre de Gabor est donnée par la relation suivante :

$$\text{T.F } g(t, \omega) = \pi^{-\frac{1}{4}} \int_{-\infty}^{+\infty} f(\tau) e^{-\frac{(\tau-t)^2}{2}} e^{-j\omega\tau} d\tau \quad (\text{III.4})$$



Fig.III.4 : représentation fréquentielle de fenêtre de Gabor.

La transformée de Gabor a été d'une grande utilité pour le traitement du signal mais elle représente un inconvénient dû aux dimensions fixes en temps et en fréquences de la fenêtre de Gabor. En effet, cette fenêtre n'est pas assez large pour les basses fréquences ni assez courte pour les hautes fréquences. Elle ne permet donc pas de prendre en considération toutes les irrégularités du signal. Afin de remédier à ce

problème, une nouvelle transformée appelée transformée en ondelettes a été proposée.

3. Définition de la transformée en ondelettes

La notion d'Ondelettes ou Wavelets (en anglais) a été utilisée pour la première fois au début des années 80 par le géophysicien français J.Morlet [10] pour désigner des fonctions mathématiques utilisées dans la représentation de données sismiques.

Les Ondelettes sont des fonctions de base de variation multi-échelles ou multirésolution, dans le but de l'approximation et/ou de la compression des données. La théorie des Ondelettes n'est pas en fait tout à fait nouvelle, mais le support mathématique, contenu déjà dans les recherches des années 30, a été développé considérablement dans les dernières 20 années, après que ces fonctions ont été utilisées dans le traitement numérique du signal et en particulier dans l'analyse et la compression des images.

L'analyse d'un signal par ondelette est réalisée par une fonction analysante spécifique Ψ appelée ondelette mère (Fig.III.5).cette dernière est positionnée dans le domaine temporel afin de sélectionner la partie du signal à traiter, elle peut être contractée ou dilatée afin de se concentrer sur une gamme de fréquence, elle peut aussi être translatée pour couvrir tout le domaine de définition du signal traité [4].

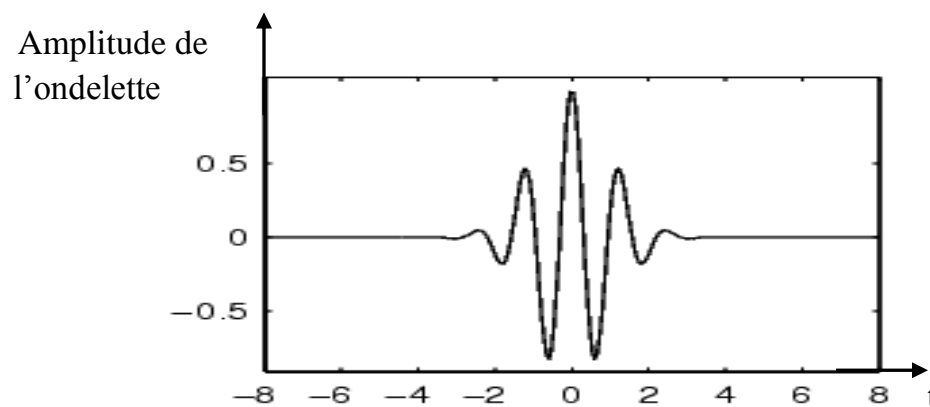
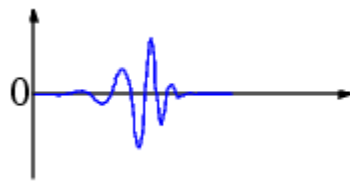


Fig. III.5: Exemple d'ondelette mère

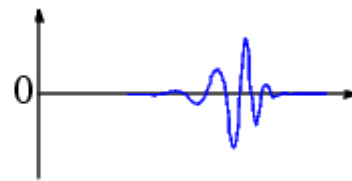
L'ondelette peut engendrer une famille $\Psi_{a,b}(t)$ ($a>0$ et b réel) par dilatation (paramètre a) (Fig.II.7) et par translation dans le temps (paramètre b) (Fig.II.6). Cette famille d'ondelette est donnée par la relation suivante :

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) \tag{III.5}$$

Le facteur de normalisation $1 / \sqrt{a}$ peut être aussi pris égal à 1 ou à $1/a$

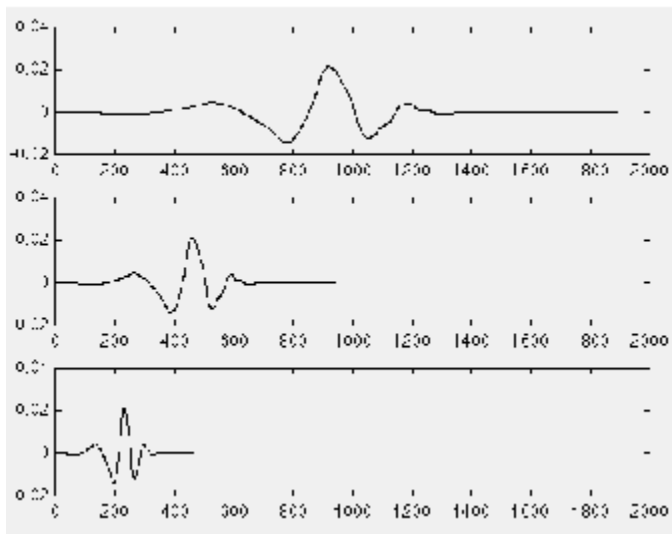


Ondelette $\Psi(t)$



Ondelette $\Psi(t-k)$ (translation de $\Psi(t)$)

Fig.III.6: Translation de l'ondelette [9]



$$f(t) = \psi(t) ; a = 1$$

$$f(t) = \psi(2t) ; a = \frac{1}{2}$$

$$f(t) = \psi(4t) ; a = \frac{1}{4}$$

Fig.III.7 : Dilatation de l'ondelette $\Psi(t)$ [9]

Une ondelette $\Psi(t)$ est une fonction admettant une décroissance suffisante, qui signifie en pratique qu'elle a au moins quelques oscillations. Cela est vrai si Ψ satisfait une condition appelée condition d'admissibilité qui est donnée comme suit :

$$C_{\Psi} = \int_0^{+\infty} \frac{|\hat{\Psi}(\omega)|^2}{\omega} d\omega < +\infty \quad [10] \quad (\text{III.6})$$

Où : $\hat{\Psi}(\omega)$ est la TF de $\Psi(t)$

L'ondelette mère doit aussi vérifier la condition d'oscillation suivante :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad [10] \quad (\text{III.7})$$

4. Transformée en ondelette continue (CWT) [6]

Le signal est un rapport physique de deux sorte d'information (temporelle et fréquentielle), chacune est définie dans son propre domaine.

Pour faire apparaître simultanément ces deux sortes d'information, J.Morlet et A.Gossman, proposent une nouvelle représentation du signal basée sur le concept de l'approche temps-échelle. Cette représentation consiste à décomposer le signal sur une famille de fonctions (ondelettes) concentrées à la fois en temps et en fréquence et de taille variable sachant que La taille de l'ondelette est inversement proportionnelle à la fréquence.

La transformée en ondelette continue est définie par la relation suivante :

$$C(a, b) = \int_{-\infty}^{+\infty} f(t) \overline{\Psi_{a,b}(t)} dt \quad (\text{III.8})$$

$\overline{\Psi_{a,b}(t)}$ est le complexe conjugué de $\Psi_{a,b}(t)$

La Transformée en ondelette inverse continue (iCWT) est donnée par La formule de reconstruction suivante :

$$f(t) = 1/C_{\Psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C(a, b) \Psi_{a,b}(t) \frac{da db}{a^2} \quad (\text{III.9})$$

5. Transformée en ondelettes discrète [6]

Dans la pratique, on préfère écrire le signal à analyser $f(t)$ comme une somme discrète plutôt qu'une intégrale. Pour cela nous discrétisons les paramètres a et b .

$$\begin{cases} a = a_0^j \\ \text{et} \\ b = k b_0 a_0^j \end{cases} \quad (\text{III.10})$$

Avec $(j, k) \in \mathbb{Z}^2$, $a_0 > 1$ et $b_0 > 0$

Dans beaucoup d'applications, il est préférable de réduire au maximum la redondance de cette représentation, pour cela, Y.Meyer (1985) a construit une ondelette particulière, qui pour des paramètres $a_0 = 2$ et $b_0 = 1$ l'ensemble des valeurs discrètes $\{(2^j, 2^j k) : j, k \in \mathbb{Z}\}$ engendre une famille d'ondelette orthogonale dans $L^2(\mathbb{R})$: ainsi en remplaçant les valeurs de a et b dans l'équation III.5 on obtient :

$$\Psi_{j,k}(t) = 2^{-j/2} \Psi(2^{-j}t - k) : j, k \in \mathbb{Z} \quad (\text{III.11})$$

La formule de reconstruction prend la forme discrète suivante :

$$f(n) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} C_{j,k} \Psi_{j,k}(n) \quad (\text{III.12})$$

Avec $(j, k) \in \mathbb{Z}^2$ j : est le paramètre d'échelle.

K : est le paramètre de translation

$$C_{j,k} = \langle f, \Psi_{j,k} \rangle = \int_{-\infty}^{+\infty} \Psi_{j,k}(t) f(t) dt$$

6. Analyse multirésolution

L'analyse multirésolution consiste à représenter un signal comme une limite de ses approximations successives, où chaque approximation est une version lissée de la précédente. Les approximations successives sont présentées à différentes résolutions d'où le nom multirésolution. Il suffit alors de passer d'un espace à un autre pour changer de résolution et modifier ainsi la taille de l'objet manipulé.

6.1 Analyse multirésolution à une dimension

Une analyse multirésolution est définie à partir de deux fonctions très liées $\Psi(t)$ et $\phi(t)$.

- La fonction $\phi(t)$, appelé fonction échelle, est définie par :

$$\phi(t) = \sqrt{2} \sum_k g(k) \phi(2t - k) \quad (\text{III.13})$$

Où $g(k)$ est une suite $\in L^2(\mathbb{R})$

- L'expression de la dilatation de $\phi(t)$ est donnée par :

$$\phi_{j,k} = 2^{-j/2} \phi(2^{-j}t - k) \quad (\text{III.14})$$

- La fonction ondelette $\Psi(t)$ est liée à la fonction $\phi(t)$ telle que :

$$\Psi(t) = \sqrt{2} \sum_k h(k) \phi(2t - k) \quad (\text{III.15})$$

$h(k)_{k \in \mathbb{Z}}$ est une suite $\in L^2(\mathbb{R})$ telle que les fonctions $\Psi_{j,k}$ pour tout j sont définies par :

$$\Psi_{j,k} = 2^{-j/2} \Psi(2^{-j}t - k) \quad (\text{III.16})$$

6.1.1 Analyse multirésolution et filtres numériques

Avant d'établir les relations existantes entre la fonction échelle et la fonction ondelette et les filtres numériques, on doit tout d'abord apporter quelques définitions.

Soit $\phi(x)$ la fonction échelle associée à une analyse multirésolution et H un filtre discret (passe-bas) de réponse impulsionnelle h définie comme suit :

$$h(n) = 2^{-1/2} \int_{-\infty}^{+\infty} \phi\left(\frac{1}{2}t\right) \phi(t - n) \quad (\text{III.17})$$

$$H(\omega) = \sum_{-\infty}^{+\infty} h(n) e^{-in\omega} \quad (\text{III.18})$$

$H(\omega)$ doit satisfaire les propriétés suivantes

$$H(0) = 1$$

$$H(\omega) \neq 0 \quad \forall \omega \in [0, \frac{\pi}{2}]$$

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 1$$

Alors la fonction $\hat{\phi}(\omega)$ est définie par

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} H(2^{-p}\omega) \quad (\text{III.19})$$

$\hat{\phi}(\omega)$ est la transformée de Fourier de la fonction d'échelle $\phi(t)$

De la même façon, soit $\Psi(t)$ la fonction ondelette associée à une analyse multirésolution et G un filtre discret (passe-haut) de réponse impulsionnelle g définie comme suit :

$$g(n) = 2^{-1/2} \int_{-\infty}^{+\infty} \Psi\left(\frac{1}{2}x\right) \phi(t - n) dt \quad (\text{III.20})$$

$$G(\omega) = \sum_{n=-\infty}^{+\infty} g(n) e^{-in\omega} \quad (\text{III.21})$$

L'ondelette mère est la fonction dont la transformée de Fourier est donnée par :

$$\hat{\Psi}(2\omega) = \hat{\phi}(\omega) e^{-i\omega} \bar{H}(\omega + \pi) \quad (\text{III.22})$$

$$\hat{\Psi}(\omega) = G\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (\text{III.23})$$

D'après les relations III.18, III.22 et III.23 nous allons avoir :

$$G(\omega) = \frac{\hat{\Psi}(2\omega)}{\hat{\phi}(\omega)} = e^{-i\omega} \sum_{n=-\infty}^{+\infty} h(n) e^{in(\omega+\pi)} \quad (\text{III.24})$$

En identifiant la relation (III.21) avec la relation (III.24) on obtient :

$$g(n) = (-1)^n h(1 - n) \quad (\text{III.25})$$

- $|G(0)| = |H(\pi)| = 0$ et $|G(\pi)| = |H(0)|$
- $|G(\omega)|^2 + |G(\omega + \pi)|^2 = 1$
- $\sum_n g(n) = 0$

Les filtres H et G sont appelé des filtres conjugués en quadrature.

6.1.2 Algorithme de transformation en coefficients d'ondelettes discrète et de reconstruction

a) Algorithme d'analyse (décomposition)

L'approximation d'une fonction $f(t)$ à la résolution 2^{-j} peut s'écrire :

$$c_j(t) = \sum_{-\infty}^{+\infty} C_j(k) \phi_{j,k}(x) \quad (\text{III.26})$$

$$c_j(k) = \langle f, \phi_{j,k} \rangle = \int_{-\infty}^{+\infty} f(u) \left(\sum_{l=-\infty}^{+\infty} h(l-2k) \phi_{j+1,l}(u) \right) du \quad (\text{III.27})$$

$$c_j(k) = \sum_{l=-\infty}^{+\infty} h(l-k) \int_{-\infty}^{+\infty} f(u) \phi_{j+1,l}(u) du = \sum_{l=-\infty}^{+\infty} h(l-k) \langle f, \phi_{j+1,l} \rangle \quad (\text{III.28})$$

$$c_j(k) = \sum_{l=-\infty}^{+\infty} h(l-2k) C_{j+1}(l) \quad (\text{III.29})$$

$$c_{j+1}(t) = \sum_{k=-\infty}^{+\infty} c_{j+1}(k) \phi_{j+1,k}(t) \quad (\text{III.30})$$

$$d_j(x) = c_{j+1}(x) - C_j(x) = \sum_{k=-\infty}^{+\infty} d_j(k) \Psi_{j,k}(x) \quad (\text{III.31})$$

$$d_j(k) = \langle f, \Psi_{j,k} \rangle = \int_{-\infty}^{+\infty} f(u) \left(\sum_{l=-\infty}^{+\infty} g(l-k) \phi_{j+1,l}(u) \right) du \quad (\text{III.32})$$

$$\begin{aligned} d_j(k) &= \sum_{l=-\infty}^{+\infty} g(l-2k) \int_{-\infty}^{+\infty} f(u) \phi_{j+1,l}(u) du \\ &= \sum_{l=-\infty}^{+\infty} g(l-2k) \langle f, \phi_{j+1,l} \rangle \end{aligned} \quad (\text{III.33})$$

$$d_j(k) = \sum_{l=-\infty}^{+\infty} g(l-2k) c_{j+1}(l) \quad (\text{III.34})$$

Les deux relations suivantes :

$$\begin{cases} c_j(k) = \sum_{l=-\infty}^{+\infty} h(l) c_{j+1}(l+2k) & (\text{III.35}) \\ d_j(k) = \sum_{l=-\infty}^{+\infty} g(l) c_{j+1}(l+2k) & (\text{III.36}) \end{cases}$$

définissent un algorithme de calcul des coefficients d'approximation et de détail à la résolution 2^{-j} d'une fonction à partir de son approximation à la résolution 2^{-j-1} (Fig.III.7).

La décomposition est équivalente à l'application d'un banc de deux filtres H^* (passe-bas) et G^* (passe-haut) tels que :

$$\left. \begin{aligned} h^*(k) &= h(-k) \\ g^*(k) &= g(-k) \end{aligned} \right\} \quad (III.37)$$

(Les H^* et G^* sont appelés filtres miroirs, respectivement, de H et G) suivie d'une décimation (on garde un échantillon sur deux).

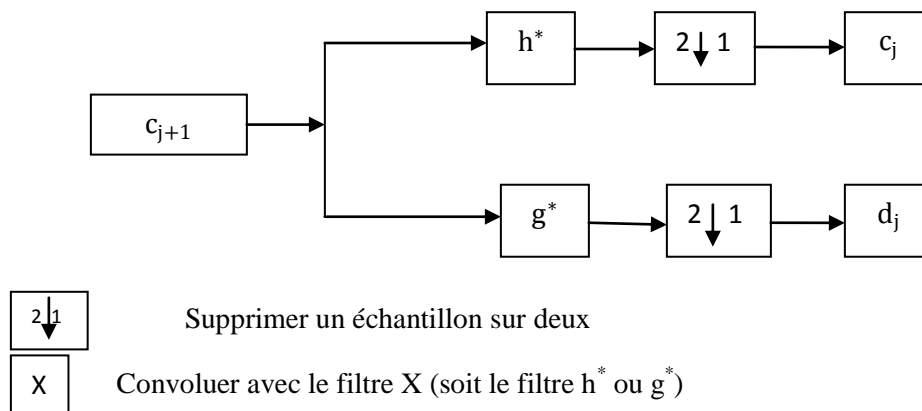


Fig.III.8: Schéma de l'analyse

b) Algorithme de reconstruction (la synthèse)

La transformée en ondelette inverse permet de reconstruire parfaitement le signal. On peut déterminer le signal à la résolution $(j+1)$ à partir du signal approximation C_j et détail d_j de la manière suivante :

$$c_{j+1}(k) = 2 \sum_{n=-\infty}^{+\infty} h(k - 2n) C_j(n) + 2 \sum_{n=-\infty}^{+\infty} g(k - 2n) d_j(n) \quad (III.38)$$

Cette reconstruction est schématisée par la figure III.9 :

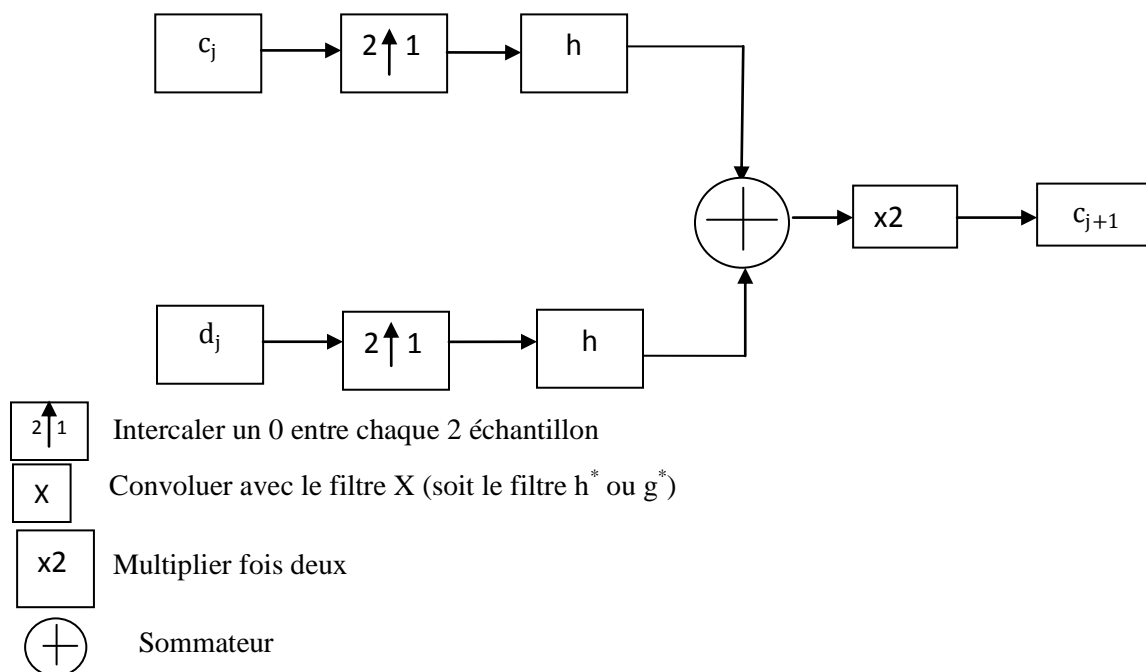


Fig. III.9: Schéma de reconstruction (synthèse).

Ainsi chaque coefficient de l'approximation est obtenu par la somme du produit des coefficients du filtre H par les coefficients de l'approximation précédente, auxquels sont intercalés des zéros et du filtrage par G des coefficients de détails auxquels sont également intercalés des zéros.

6.1.3 Illustration par un exemple numérique

Supposons que nous avons un signal défini sur l'intervalle $[0,1]$. Pour avoir une approximation discrète du signal nous pouvons prendre ses valeurs dans deux points, quatre points, huit points et ainsi de suite, le diviser en deux fonctions, de 0 à $1/2$ et de $1/2$ à 1, puis en quatre fonctions, de 0 à $1/4$, de $1/4$ à $1/2$, de $1/2$ à $3/4$, et de $3/4$ à 1 etc... On obtient le signal avec différentes résolutions et pour chaque résolution on peut avoir une représentation dans l'espace des fonctions à l'aide d'un système de fonctions de base, nommées fonctions de base multi - résolutions ou multi - échelles. Les ondelettes sont des fonctions de base multi - échelles qui assurent le passage cohérent entre les différentes résolutions, la décomposition et la reconstitution de la fonction

représentée. Si on utilise les ondelettes comme système de fonctions de base, à chaque niveau on dispose des approximations (moyennes) de la fonction initiale et des informations de détails, ou différences.

Dans la figure III.10, on a un exemple de fonction et de sa représentation au niveau I à l'aide des ondelettes de Haar. Les ondelettes de Haar sont les plus simples et les premières à avoir été utilisées.

La transformée de Haar de la fonction $f(x)=[y_1 \ y_2 \ y_3 \dots y_n]$ génère:

- Des approximations $[a_1 \ a_2 \ a_3 \dots a_{n/2}]$ qui sont les moyennes des valeurs initiales de la fonction prises deux par deux: $a_1=(y_1+y_2)/2 \dots$
- des coefficients de détail ou les différences, $[d_1 \ d_2 \dots d_{n/2}]$, avec $d_1=y_1-a_1$, $d_2=y_3-a_2 \dots$

Ce processus est équivalent à une convolution de la fonction avec le filtre bas $g=[0.5 \ 0.5]$ et, pour obtenir les coefficients de détail, nécessaires à la reconstruction, il faut effectuer la transformée inverse.

Toutes ces étapes sont clarifiées par les deux figures qui suivent.

La figure III.11 illustre la fonction d'échelle et l'ondelette mère ainsi que les différentes ondelettes engendrées par translation ou dilatation de l'ondelette mère .

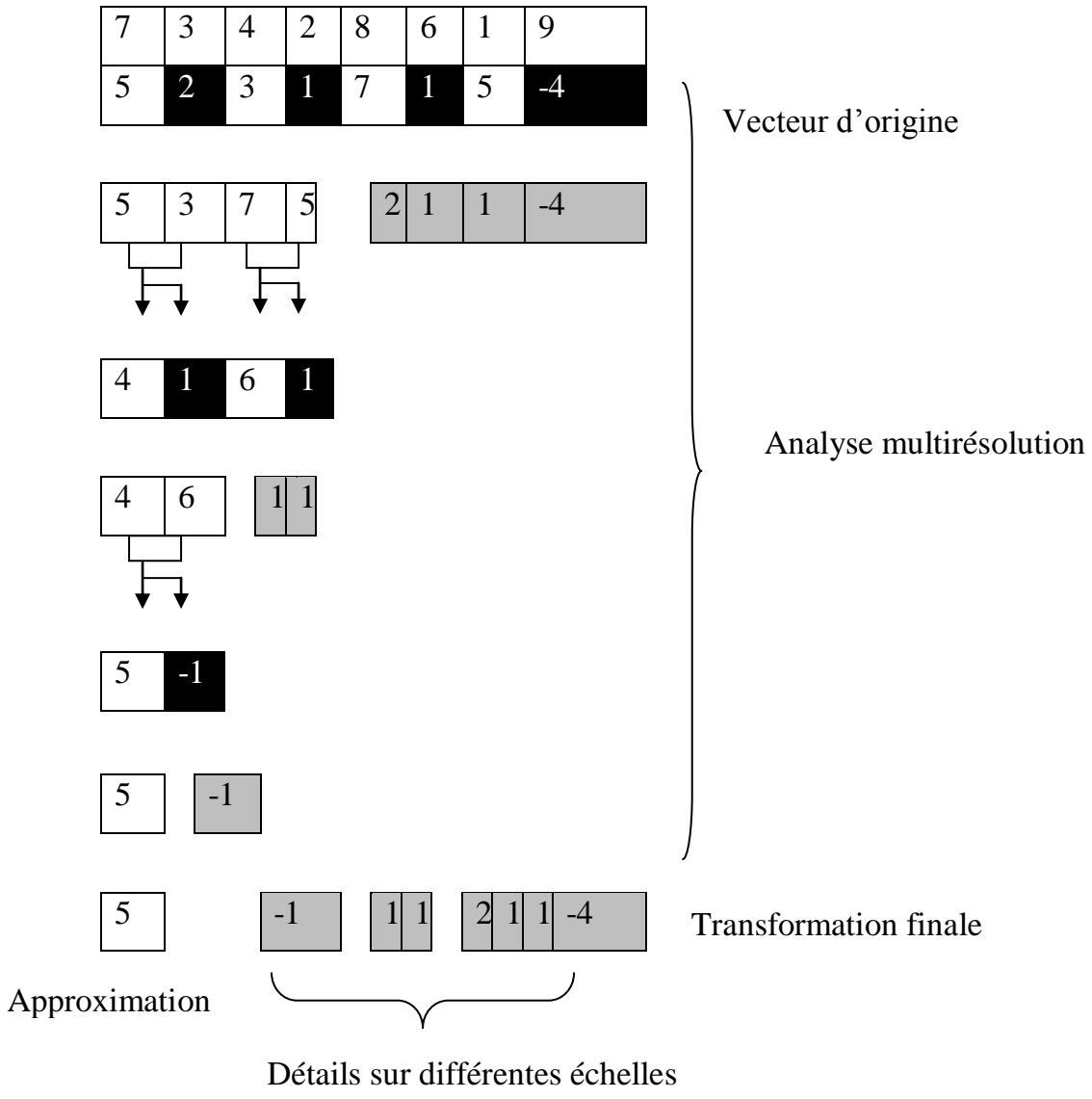


Fig.III.10: Exemple d'analyse multirésolution

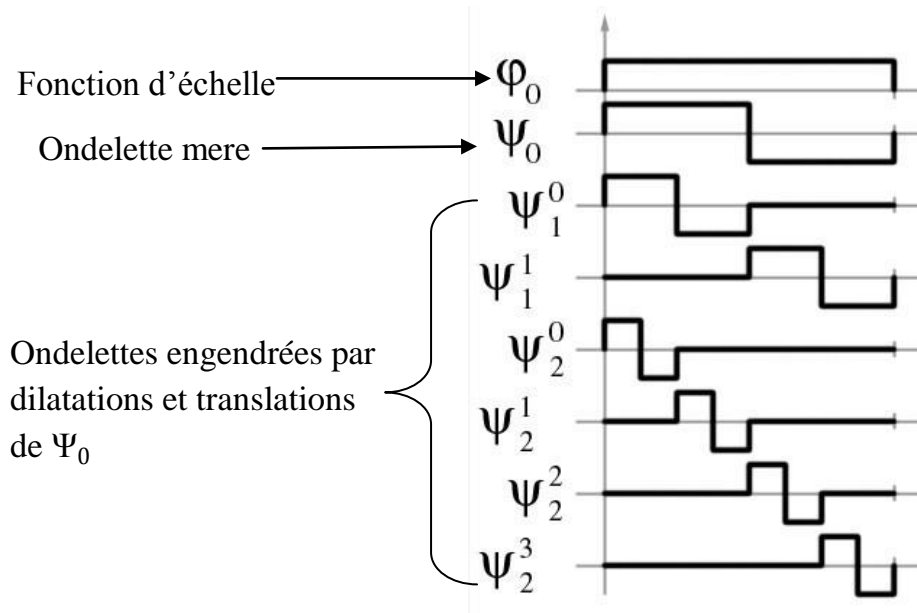


Fig.III.11: la fonction d'échelle et les ondelettes utilisées dans l'analyse de Haar

L'opérateur de reconstruction, est dans ce cas très simple :

$$y_1 = a_1 + d_1 ; y_2 = a_1 - d_1 ; y_3 = a_2 + d_2 ; y_4 = a_2 - d_2, \text{ et ainsi de suite.}$$

En général, si on note $a_0(t)$ les valeurs de $f(x)$ alors les coefficients d'approximations et de détails sont donnés par les formules:

$$a_{m+1}(t) = a_m(2t) + a_m(2t+1) \quad (39)$$

$$d_{m+1}(t) = a_m(2t) - a_m(2t+1), \text{ pour } m \geq 0 \quad (40)$$

6.2 Analyse multirésolution 2D

S.Mallat a exploité les propriétés, très intéressantes, de l'analyse multirésolution et de la théorie des ondelettes pour construire des algorithmes de décomposition en coefficients d'ondelettes discrètes et de reconstruction des images 2- dimensions.

Pour appliquer le concept introduit en une dimension au cas bidimensionnel, et notamment aux images, le principe le plus utilisé est le produit tensoriel des opérateurs définis en une dimension tel que :

Soit $\Psi(x)$ la fonction ondelette associée la fonction échelle $\phi(x)$. On peut construire les 4 fonctions suivantes :

$$\left. \begin{aligned} \phi(x,y) &= \phi(x)\phi(y) \\ \Psi^1(x,y) &= \phi(x)\Psi(y) \\ \Psi^2(x,y) &= \Psi(x)\phi(y) \\ \Psi^3(x,y) &= \Psi(x)\Psi(y) \end{aligned} \right\} [4] \quad (\text{III.41})$$

Par analogie au cas à une dimension on a alors :

$$\Phi_{jk}(x,y) = 2^{-j} \Phi_{jk}(2^{-j}x - k_1, 2^{-j}y - k_2) = \phi_{jk_1}(x) \phi_{jk_2}(y) \quad (\text{III.42})$$

$$\Psi^1_{jk}(x,y) = \phi_{jk_1}(x) \Psi_{jk_2}(y) = 2^{-j} \phi(2^{-j}x - k_1) \Psi(2^{-j}y - k_2) \quad (\text{III.43})$$

$$\Psi^2_{jk}(x,y) = \Psi_{jk_1}(x) \phi_{jk_2}(y) = 2^{-j} \Psi(2^{-j}x - k_1) \phi(2^{-j}y - k_2) \quad (\text{III.44})$$

$$\Psi^3_{jk}(x,y) = \Psi_{jk_1}(x) \Psi_{jk_2}(y) = 2^{-j} \Psi(2^{-j}x - k_1) \Psi(2^{-j}y - k_2) \quad (\text{III.45})$$

Un raisonnement analogue à celui fait en une seule dimension donne l'expression des coefficients d'approximation et de détail d'une fonction bidimensionnelle $f(x,y)$:

$$\left. \begin{aligned} c^j &= \langle f, \phi_{jk_1} \rangle = 2^j \langle f(x,y) \phi(2^j x - k, 2^j y - l) \rangle \\ d^j_v &= \langle f, \Psi^1_{jk_1} \rangle = 2^j \langle f(x,y) \Psi^1(2^j x - k, 2^j y - l) \rangle \\ d^j_H &= \langle f, \Psi^2_{jk_1} \rangle = 2^j \langle f(x,y) \Psi^2(2^j x - k, 2^j y - l) \rangle \\ d^j_D &= \langle f, \Psi^3_{jk_1} \rangle = 2^j \langle f(x,y) \Psi^3(2^j x - k, 2^j y - l) \rangle \end{aligned} \right\} \quad (\text{III.46})$$

Ces coefficients peuvent être également obtenus par filtrage en utilisant les expressions suivantes :

$$c^j = \sum_n \sum_m h(2k - n) h(2l - m) c^{j+1}_{n,m} \quad (\text{III.47})$$

$$d^j_v = \sum_n \sum_m h(2k - n) g(2l - m) c^{j+1}_{n,m} \quad (\text{III.48})$$

$$d^j_H = \sum_n \sum_m g(2k - n) h(2l - m) c^{j+1}_{n,m} \quad (\text{III.49})$$

$$d_D^j = \sum_n \sum_m g(2k - n)g(2l - m) c^{j+1}_{n,m} \tag{III.50}$$

Les schémas des figures III.12 et III.13 illustrent le principe de calcul de ces coefficients.

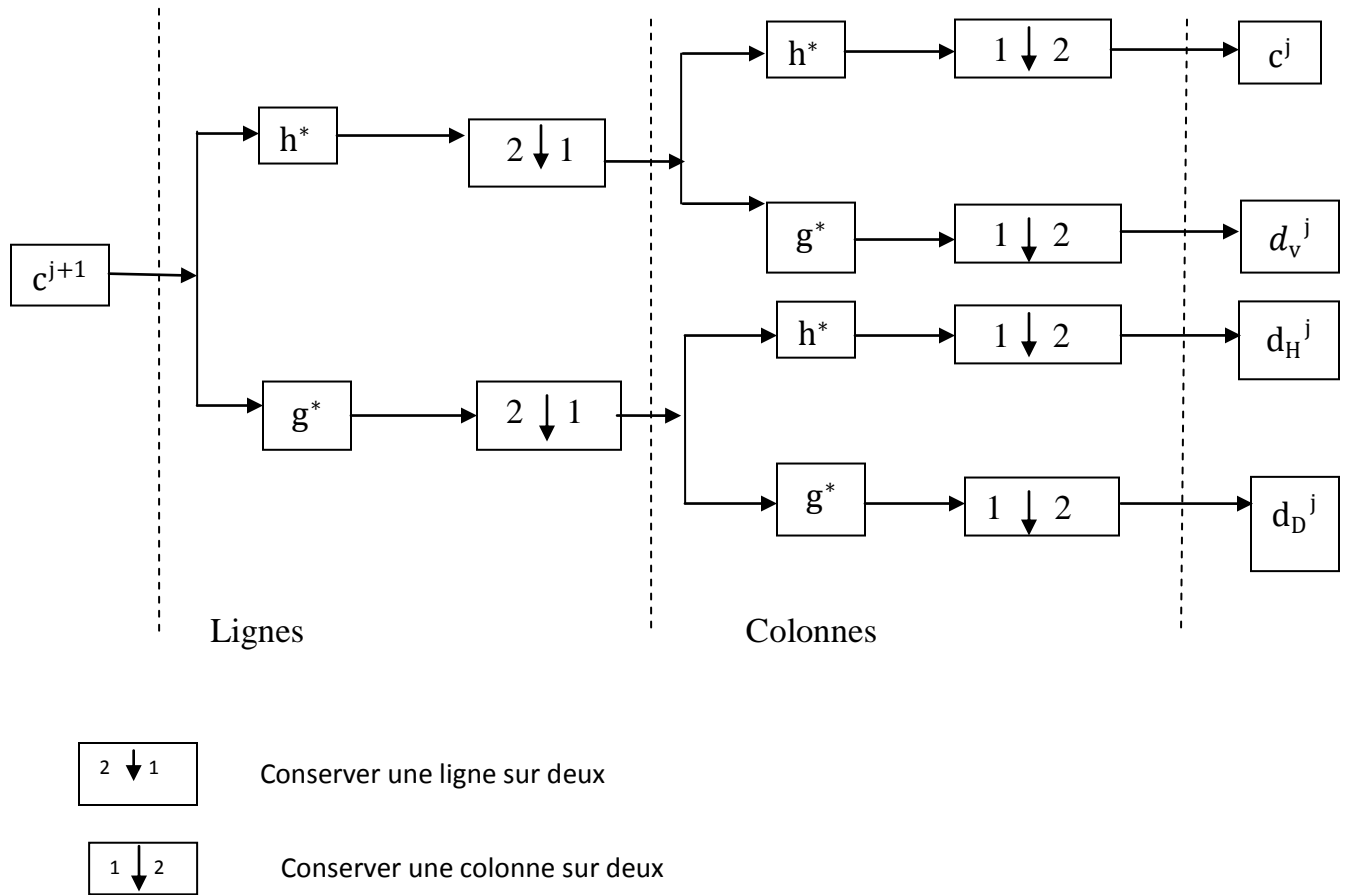


Fig.III.12: Schéma de décomposition [5]

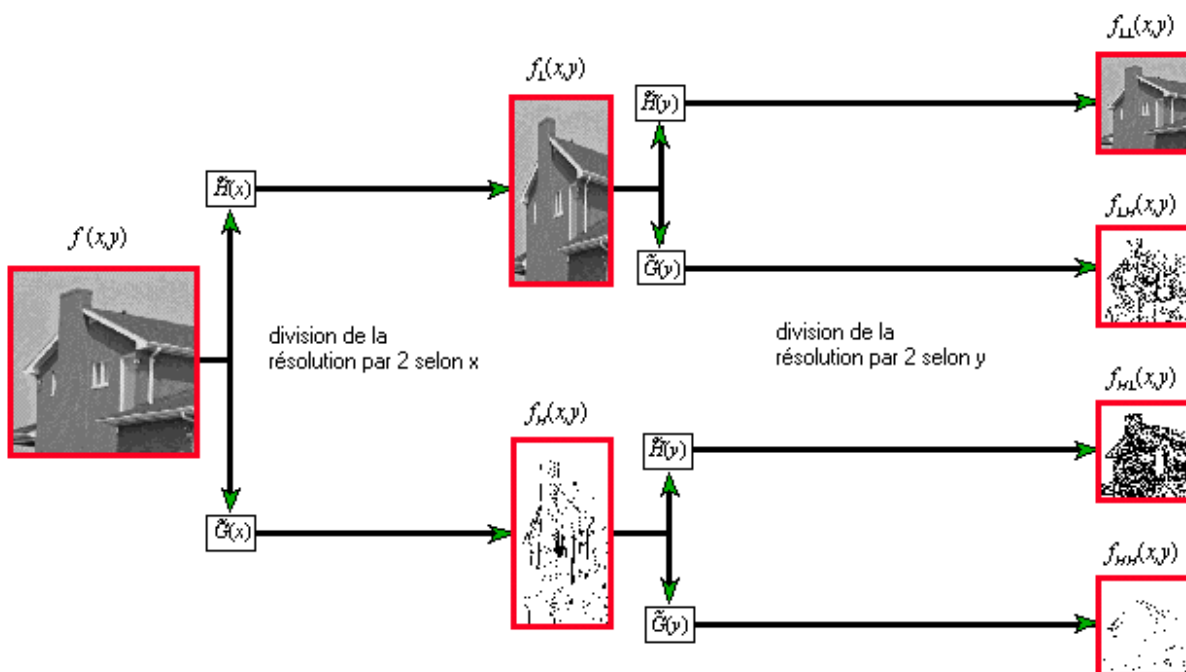


Fig.III.13:exemple de décomposition de l'image House [5]

L'application de l'algorithme de décomposition de S.MALLAT, à une résolution j sur une image initiale C^0 de taille T , consiste à calculer l'approximation C^j à la dernière résolution et un ensemble de détails $\{D^{k,m}\}$ à tous les niveaux de résolution k inférieurs à j .

À chaque itération k de l'algorithme, nous obtenons l'approximation C^k de taille $2^{-2km}T$ et les trois bandes de détails (D_V^j, D_H^j, D_D^j) de taille $2^{-2km}T$ avec $1 \leq m \leq 3$. Le total nous donne la taille initiale T . À la première étape de l'algorithme, on décompose l'image originale en C^1 par la transformée en ondelette discrète, C^1 représente l'approximation à la résolution 1 et un ensemble de détails (D_V^1, D_H^1, D_D^1) . C^1 est obtenu en appliquant un filtre-passe-bas h en prenant une colonne sur deux (décimation d'une colonne sur deux de C^0), ensuite nous appliquons le même filtre h en prenant cette fois ci une ligne sur deux de C^0 (décimation d'une ligne sur deux de C^0).

Les détails (D_V^1, D_H^1, D_D^1) traduisent l'information perdue entre C^0 et C^1 , sont

obtenus en appliquant respectivement les filtre résultants du produit tensoriel des filtres h et g en prenant soin de décimer une fois les colonnes et une fois les lignes.

De la même manière que dans le cas monodimensionnel, la reconstruction se des approximations bidimensionnelles peut se faire selon la formule suivante :

$$c^{j+1} = \sum_n \sum_m [\hat{h}(k-2n) \hat{h}(l-2m) c^j + \hat{h}(k-2n) \hat{g}(l-2m) d_v^j + \hat{g}(k-2n) \hat{h}(l-2m) d_H^j + g(k-2n) \hat{g}(l-2m) d_D^j] \quad (51)$$

Cette reconstruction est schématisée par la figure suivante :

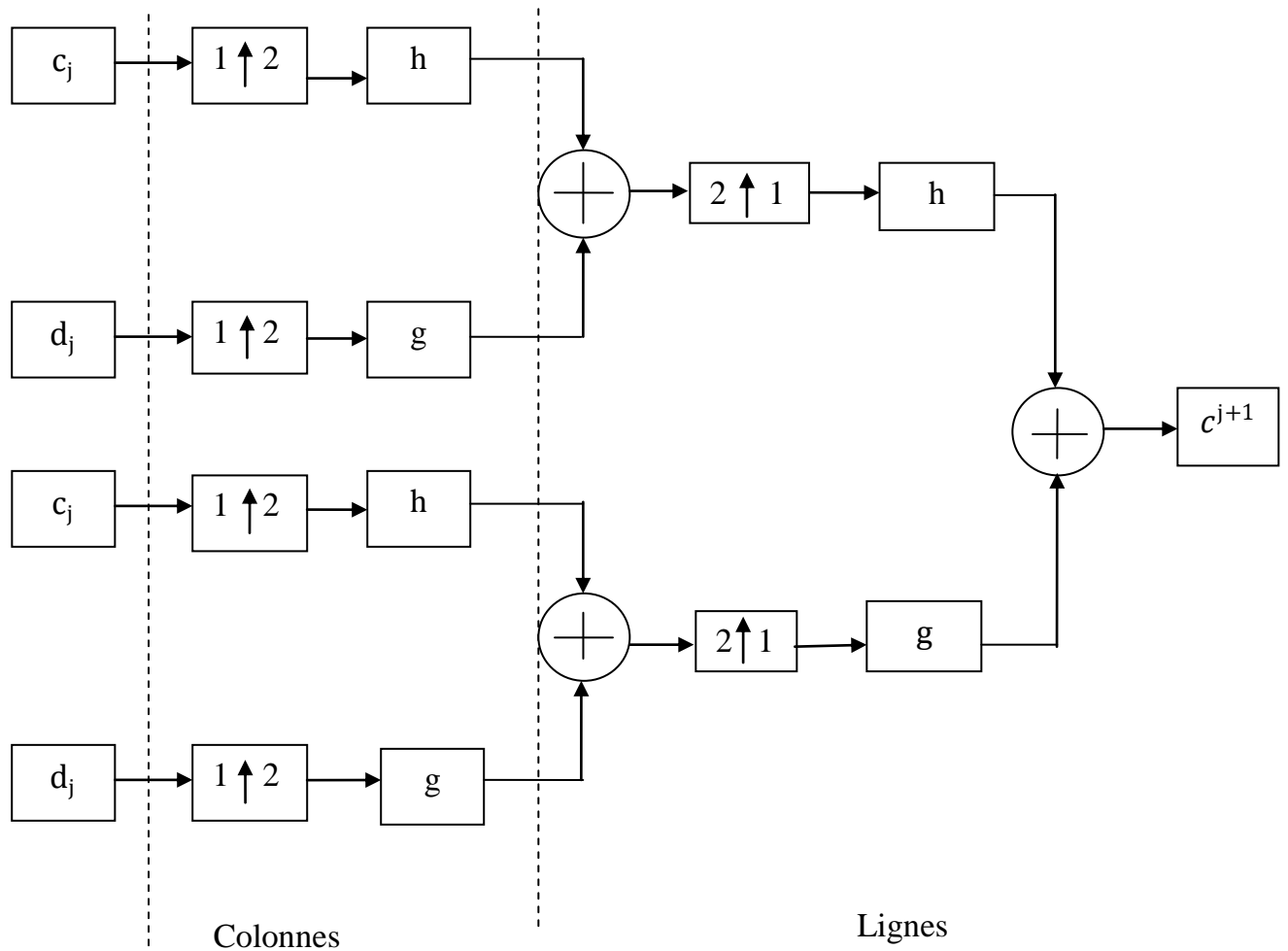


Fig.III.14: Schéma de reconstruction à deux dimensions [5]

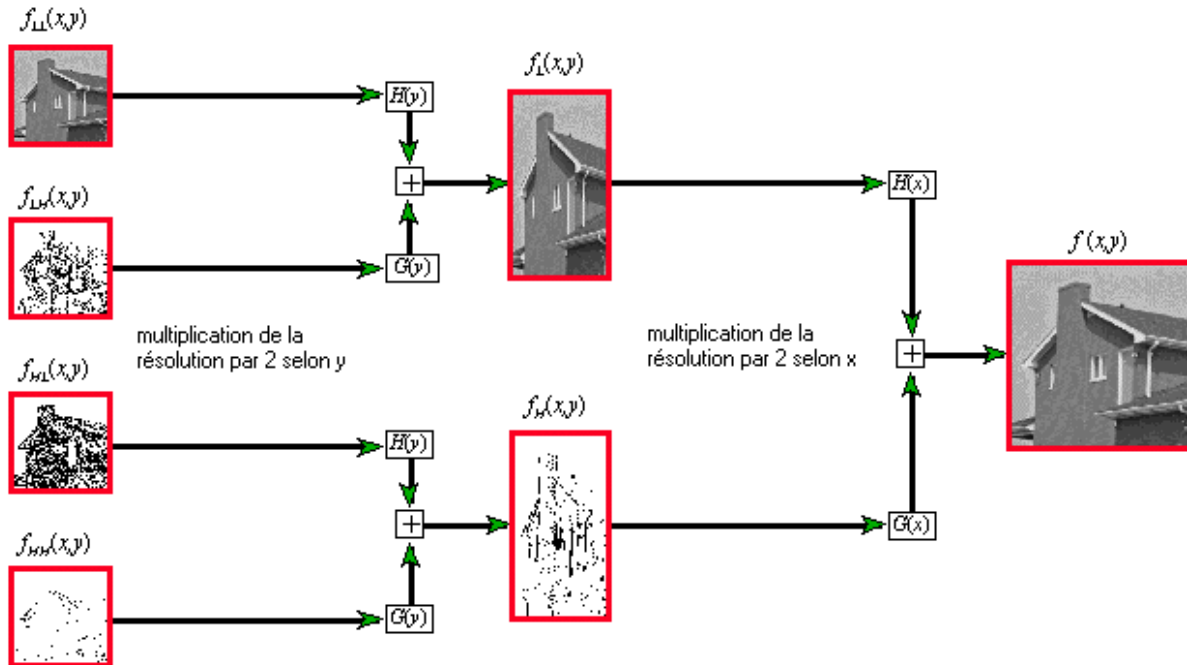


Fig.III.15:exemple de reconstruction de l'image House

7. Conclusion

A travers ce chapitre nous avons donné un bref aperçu sur la théorie des ondelettes, qui présente un moyen mathématique très puissant permettant de donner une autre représentation de l'information contenue dans un signal (image) que celle donnée par l'analyse de Fourier.

La transformée en Ondelettes permet une analyse complète du signal stationnaire ou pas, sans pertes de localité et sans l'inconvénient des largeurs des fenêtres analysantes. Il est impératif de noter que le choix d'une ondelette analysante par rapport à une autre ne peut se faire que sur la base du problème à résoudre. Dans le cas des signaux numériques l'algorithme est rapide et puissant, permet aussi des applications professionnelles comme la compression des images.

Chapitre IV

Tests et résultats

1. Introduction

Le but de ce travail est de réaliser une segmentation d'images en essayant de réduire le temps de calcul. Dans ce travail, nous allons utiliser la méthode de segmentation nommée JSEG. Cette dernière est très utilisée pour la segmentation d'images texturées. Elle est réalisée en plusieurs étapes, fait intervenir plusieurs techniques, ce qui se répercute sur le temps de calcul. Nous proposons, dans ce mémoire, de réduire ce temps en réduisant le nombre de données à traiter. Cette réduction sera réalisée en utilisant les transformées en ondelettes.

Nous avons implémenté notre programme sous MATLAB 7.10 et réalisé, au cours de ce chapitre, plusieurs tests de la méthode proposée, sur des images réelles et synthétiques.

2. Méthode proposée

La segmentation d'images par l'algorithme JSEG nécessite un temps de calcul important. Pour remédier à ce problème, nous proposons une approche en trois étapes qui sont :

- réduction du volume des données en réduisant la taille de l'image
- segmentation de l'image réduite
- agrandissement de l'image segmentée pour retrouver la taille de l'image initiale.

Dans la première étape, nous réduisons le nombre de données à traiter en appliquant la transformée en ondelettes. Par la suite, sauf l'image approximation sera prise en considération alors que les images détails seront ignorées. La décomposition peut se faire sur plusieurs niveaux (voir figure IV.1). Pour une décomposition de niveau 1, les données à traiter seront réduites de 4, alors que pour une décomposition de niveau 2, elles seront réduites de 16, et ainsi de suite, pour une décomposition de niveau n , elles sont réduites de 4^n .

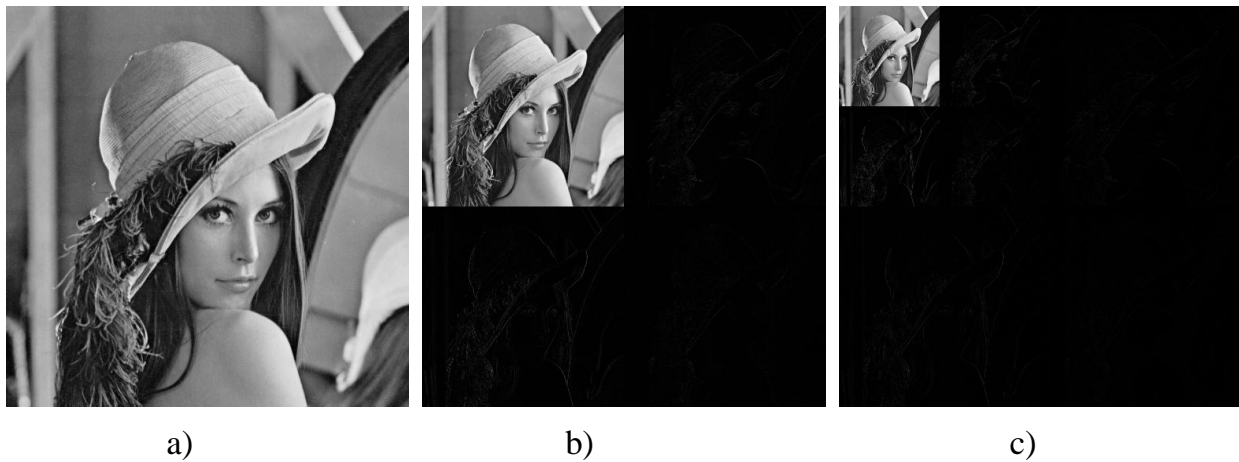


Fig.IV.1 : Décomposition de l'image Lenna par l'ondelette Haar
a) image originale b) décomposition de niveau 1 c) décomposition de niveau 2

Il existe plusieurs types d'ondelettes, dans ce travail, nous testerons les plus utilisées qui sont Haar, Daubechies et Coiflet.

La figure IV.2 fournit la décomposition de l'image Lenna en utilisant ces trois types d'ondelettes.



Fig.IV.2 : Décomposition de l'image Lenna par trois types d'ondelettes
a) ondelette Haar b) ondelette Daubechies c) ondelette Coiflet

La seconde étape de la méthode consiste à segmenter l'image approximation au niveau de décomposition choisi, en utilisant l'algorithme JSEG développé dans le Chapitre II (voir figure IV.3).

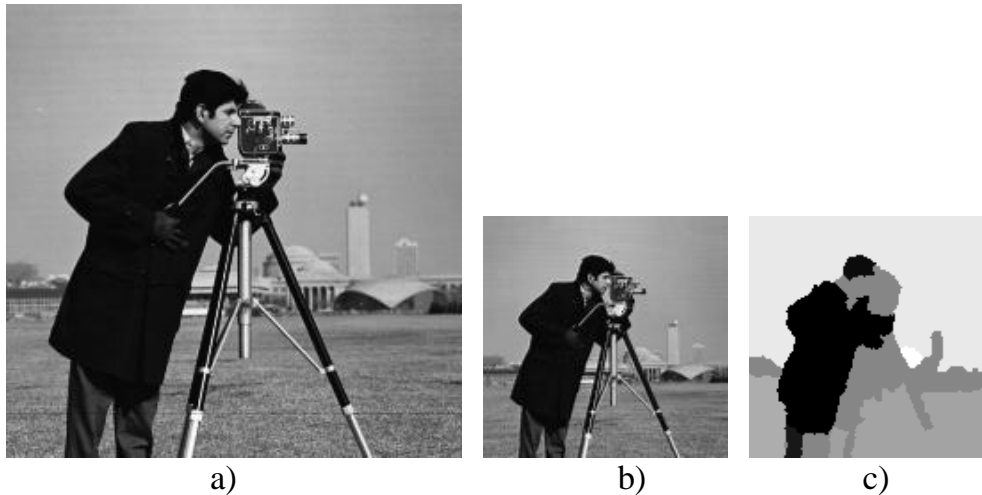


Fig. IV.3 : Segmentation de l'image approximation par la méthode JSEG
a) image originale b) image approximation par l'ondelette Daubechies c) segmentation de l'image approximation

Dans la dernière étape, l'image segmentée obtenue précédemment sera agrandie pour avoir la même taille que l'image originale. L'agrandissement sera effectué en réalisant la transformée en ondelettes inverse sans prendre en compte les détails. Un exemple illustrant cette étape est fourni sur la figure IV.4. Il est évident que la transformée en ondelettes inverse utilisée dans cette étape sera de même nature que celle utilisée dans la première étape.

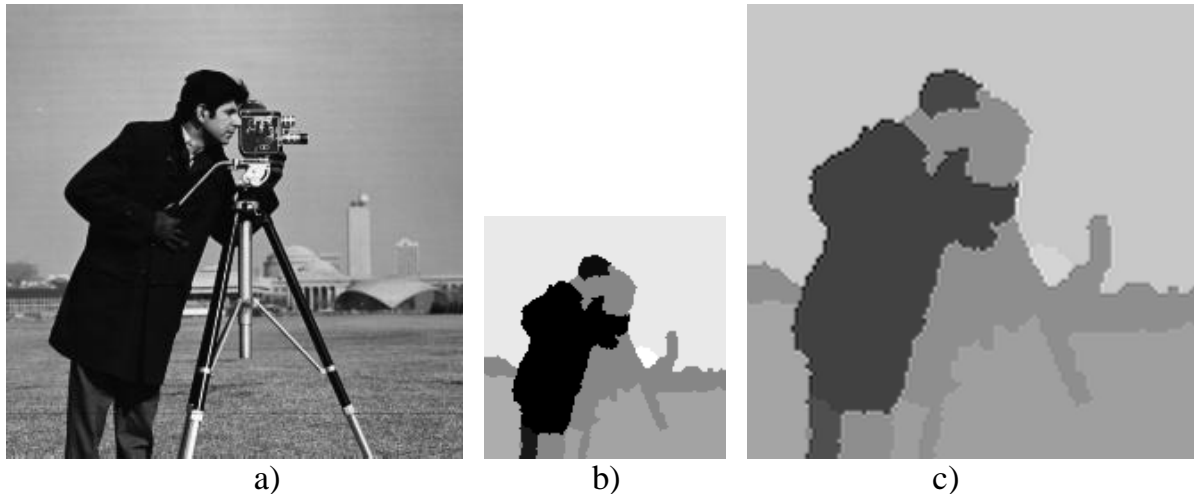


Fig. IV.4 : Agrandissement après segmentation de l'image approximation
a) image originale b) image approximation segmentée c) agrandissement de b)

Les résultats de la segmentation peuvent être illustrés de deux façons différentes. La première consiste à représenter les différentes régions avec des niveaux de gris différents elle sera appelée « carte des régions », dans la deuxième, nous superposons les contours des régions à l'image avant segmentation. Ces deux types d'illustration sont affichés sur la figure IV.5. Ne n'utiliserons par la suite que la première représentation.

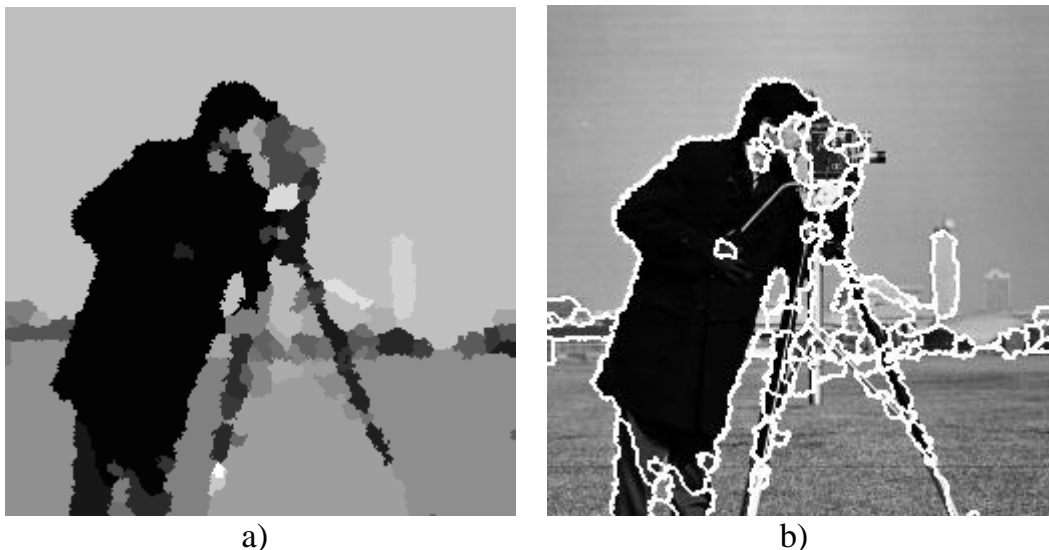


Fig. IV.5 : Représentation des résultats
a) carte des régions b) résultats affichés sur l'image originale

Pour réaliser ce travail nous devons, au préalable, fixer un certain nombre de paramètres, certains sont relatifs à la réduction de la taille de l'image et d'autres à la méthode de segmentation JSEG.

Les paramètres relatifs à la réduction de la taille de l'image sont :

- le type de l'ondelette et
- le niveau de décomposition.

Les paramètres relatifs à la méthode JSEG sont :

- la taille de la fenêtre utilisée pour le calcul des poids perceptuels $w*w$,
- le seuil de quantification QT et
- le seuil de fusion S.

Pour apprécier l'influence de chacun de ces paramètres, nous allons à chaque fois faire varier le paramètre en question et fixer les autres.

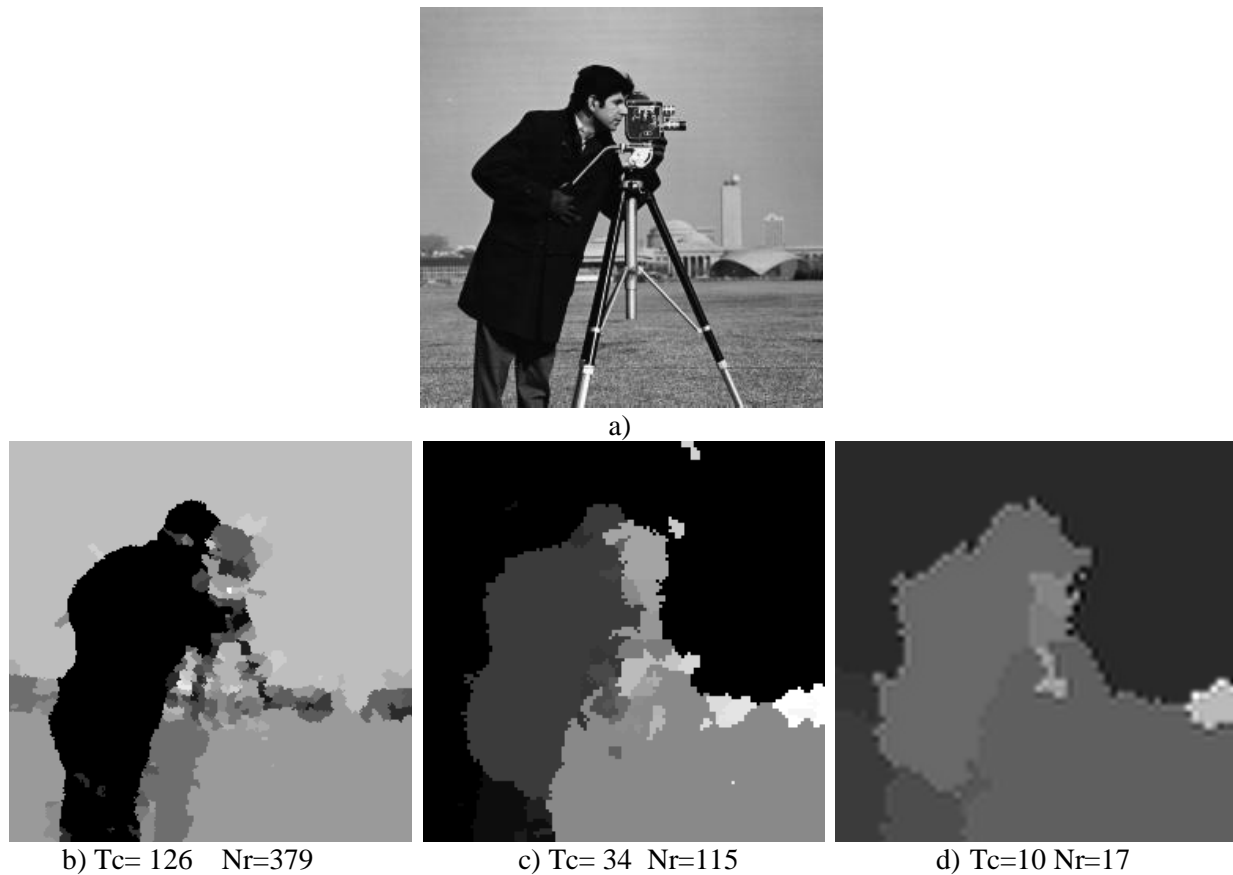
3. Influence du choix du niveau de décomposition de l'ondelette

L'application de la transformée en ondelettes et en ne considérant que l'image approximation, nous fait perdre une bonne partie de l'information. Dans ce paragraphe, nous proposons de voir si le niveau de décomposition, c'est-à-dire le nombre de fois que la transformée est appliquée, influe sur le résultat de la segmentation. Pour vérifier cela nous allons fixer les autres paramètres et comparer les résultats obtenus sans décompositions avec ceux obtenus avec des décompositions de niveau 1 et 2.

Les autres paramètres sont fixés aléatoirement comme suit :

- le type le type de l'ondelette : Haar
- la taille de la fenêtre utilisée pour le calcul des poids perceptuels $w*w=9*9$,
- le seuil de quantification QT=10 et
- le seuil de fusion S=10.

Dans les figures ci-dessous, les résultats de la segmentation, sans décomposition et avec décomposition de niveau 1 et 2, sont fournies sous forme de carte des régions. Nous afficherons à chaque fois le nombre de régions détectées N_r ainsi que temps de calcul T_c .



b) $T_c=126$ $N_r=379$

c) $T_c=34$ $N_r=115$

d) $T_c=10$ $N_r=17$

Fig. IV.6 : Résultats de la segmentation de l'image Cameraman à différents niveaux de décomposition de l'ondelette Haar

a) Image originale b) Résultat sans décomposition c) Résultat avec une décomposition de niveau 1
d) Résultat avec une décomposition de niveau 2

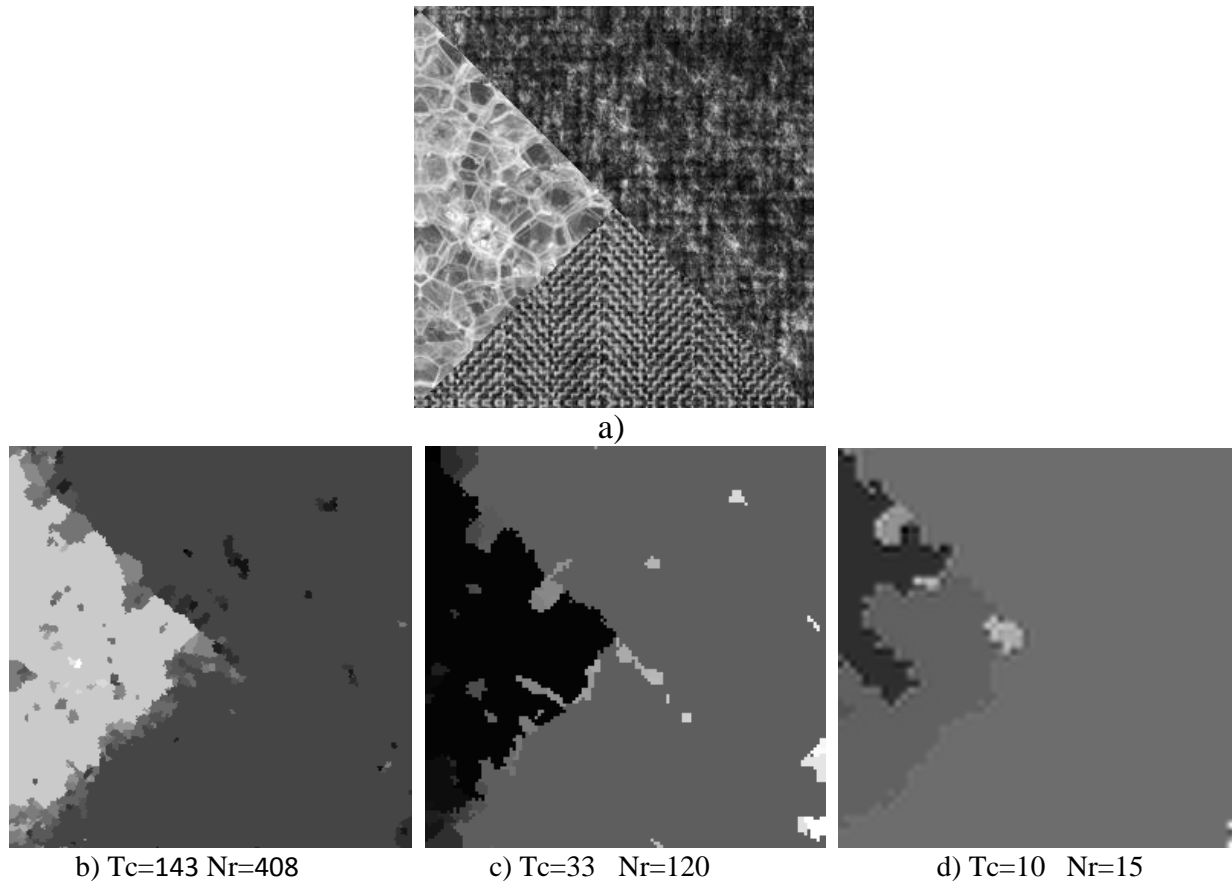


Fig. IV.7 : Résultats de la segmentation d'une image Synthétique à différents niveau de décomposition de l'ondelette Haar

a) Image originale b) Résultat sans décomposition c) Résultat avec décomposition de niveau 1 d) Résultat avec décomposition de niveau 2

Les résultats obtenus sur des images réelles, donnés en figures IV.6 montrent que plus le niveau de décomposition augmente, plus les résultats de la segmentation se détériorent. Ce résultat peut s'expliquer par le fait que nous avons éliminé les détails de l'image originale en effectuant les décompositions. En effet, rien que les images approximées qui sont segmentées. Pour ce qui concerne l'image synthétique les segmentations obtenues, sans et avec décomposition, ne sont pas concordantes avec la réalité. En effet dans les trois résultats, deux textures se confondent. Il est probablement possible que les valeurs des paramètres relatifs à la méthode JSEG sont mal choisis. Le choix de ces derniers sera étudié dans les paragraphes suivants.

Nous remarquons que lorsque, nous effectuons une segmentation sur l'image approximée au niveau 1, le temps de calcul se réduit considérablement et les différentes

régions présentes dans l'image sont globalement détectées. Pour ce qui est du niveau 2, le temps de calcul est bien sur nettement inférieur sauf que le résultat de la segmentation est mauvais.

Pour la suite, nous allons tester uniquement la décomposition de niveau 1 et comparer ses résultats avec celles sans décompositions.

4. Influence du choix du type de l'ondelette

Pour illustrer l'influence du type d'ondelettes sur le résultat final de la segmentation, nous appliquons sur l'image originale trois types d'ondelettes qui sont les ondelettes de Haar, Daubechies et Coiflet. Sur l'image obtenue après transformation, nous allons appliquer l'algorithme JSEG et puis à la fin, pour revenir à la taille initiale nous appliquerons une transformée en ondelettes inverse de même type et nous comparerons les résultats obtenus. Les valeurs des autres paramètres seront fixées comme suit :

- le niveau de décomposition de l'ondelette : 1
- le seuil de quantification $QT=10$,
- la taille de la fenêtre de balayage $w \times w=9 \times 9$ et
- le seuil de fusion $S=10$.

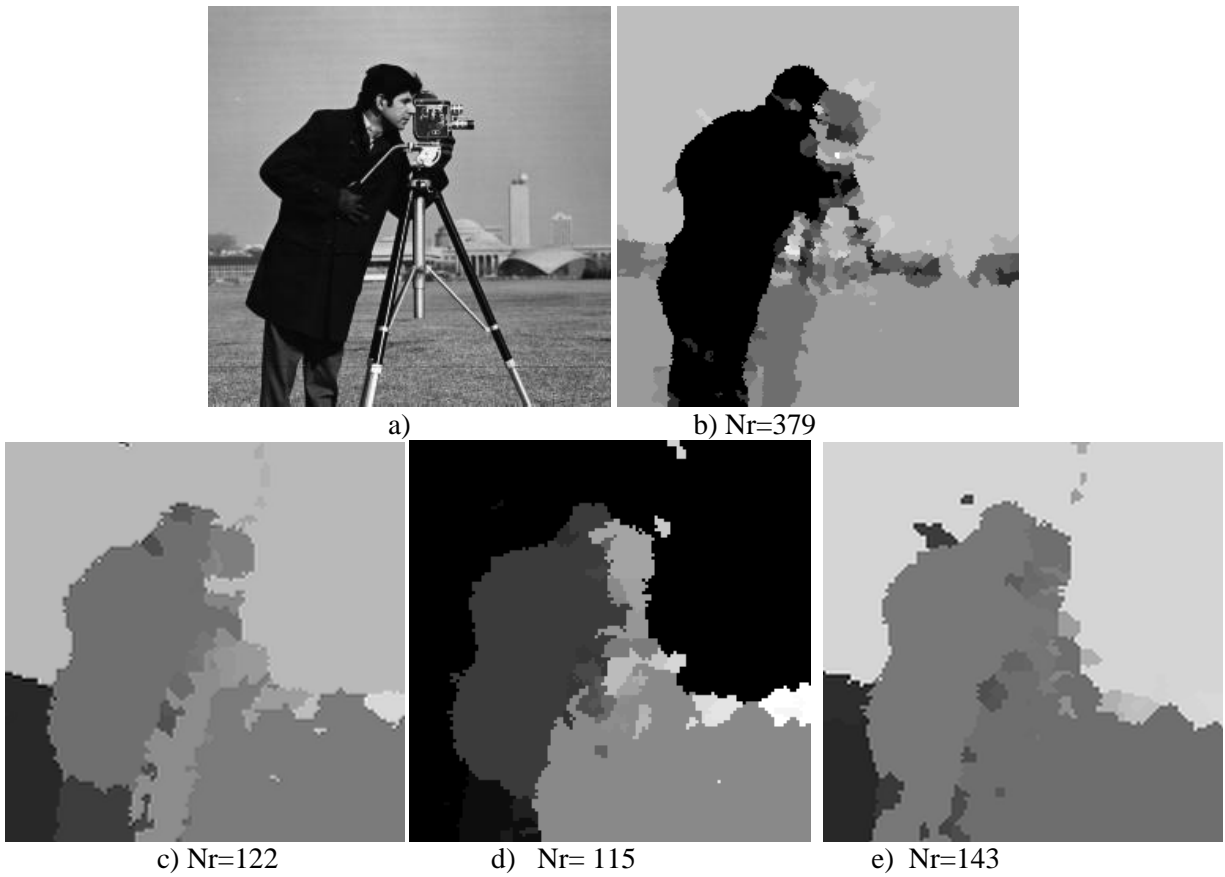


Fig. IV.8 : Résultats de la segmentation de l'image Cameraman au niveau de décomposition 1

a) Image originale b) Résultat sans décomposition
 Résultat avec décomposition de niveau 1 par l'ondelette c) Daubechies d) Haar e) Coiflet

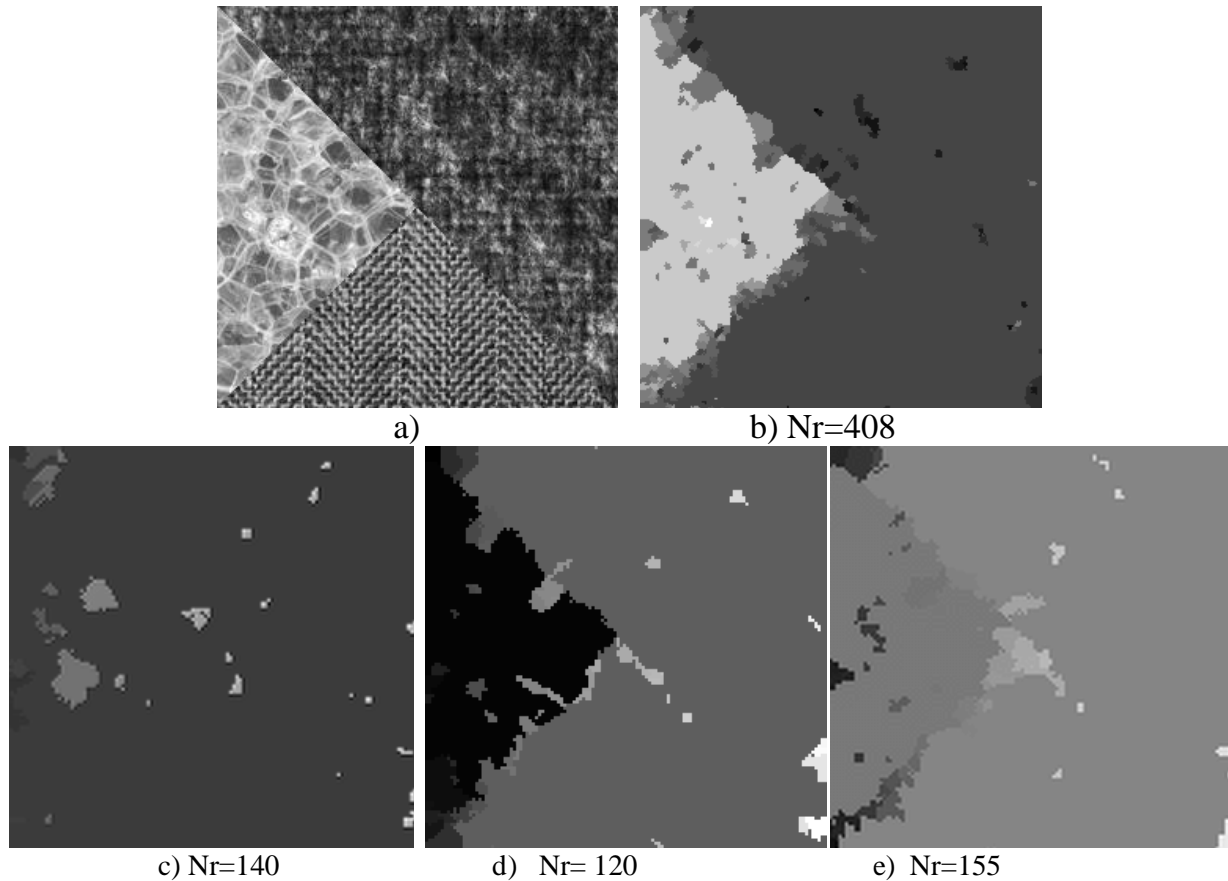


Fig. IV.9 : Résultats de la segmentation de l'image synthétique au niveau de décomposition 1

a) Image originale b) Résultat sans décomposition
 Résultat avec décomposition de niveau 1 par l'ondelette c) Daubechies d) Haar e) Coiflet

Nous observons sur les résultats affichés sur les deux figures précédentes que le choix du type de l'ondelette influe peu sur les résultats. L'utilisation de l'ondelette de Daubechies fourni un mauvais résultat sur l'image synthétique ; ceci n'est pas dû au type de l'ondelette, mais au seuil de fusion S qui n'est pas adéquat comme nous pouvons l'observer sur la figure IV.10.

Nous optons pour la suite des tests pour l'ondelette Haar car elle est souvent utilisée dans l'analyse de la texture.

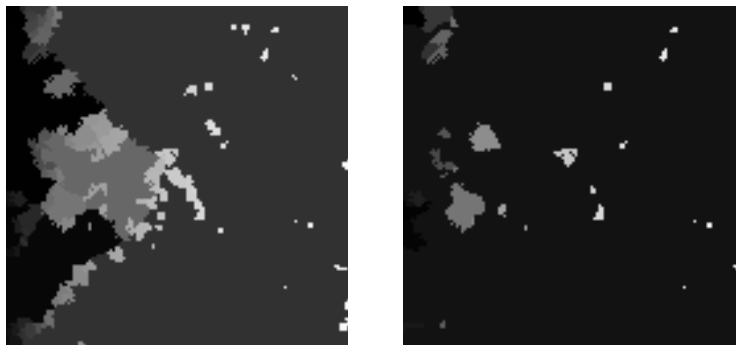


Fig. IV.10 : Résultats de la segmentation de l'image synthétique au niveau de décomposition 1 par l'ondelette de Daubechies
a) Avant fusion des régions b) Après fusion des régions avec $S=10$

5. Influence du seuil de fusion S

Dans l'étape de fusion au cours de la segmentation spatiale nous avons un paramètre S à régler au préalable, pour fusionner toutes les régions dont la distance est inférieure à ce seuil, le processus de fusion est arrêté lorsque toutes les régions voisines aient une distance supérieure à S .

De la même manière que dans les étapes précédentes, nous allons tester l'influence de ce paramètre sur différentes images, en faisant varier sa valeur S , tout en fixant les autres paramètres comme suit :

- type de l'ondelette : Haar
- niveau de décomposition de l'ondelette : 1
- seuil de quantification $QT= 10$.
- taille de la fenêtre de balayage $w \times w=9 \times 9$.

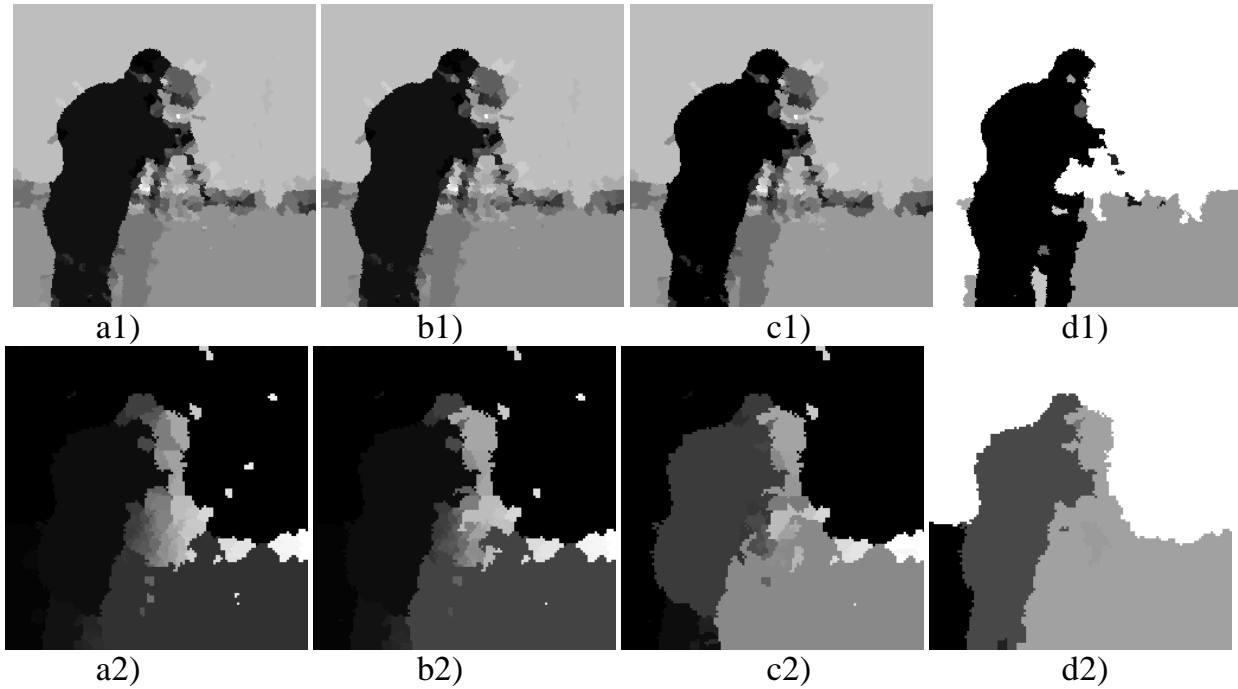


Fig. IV.11 : Résultats de la segmentation de l'image Cameraman en utilisant différents seuils de fusion

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) S=0 b) S=5 c) S=10 d) S=50

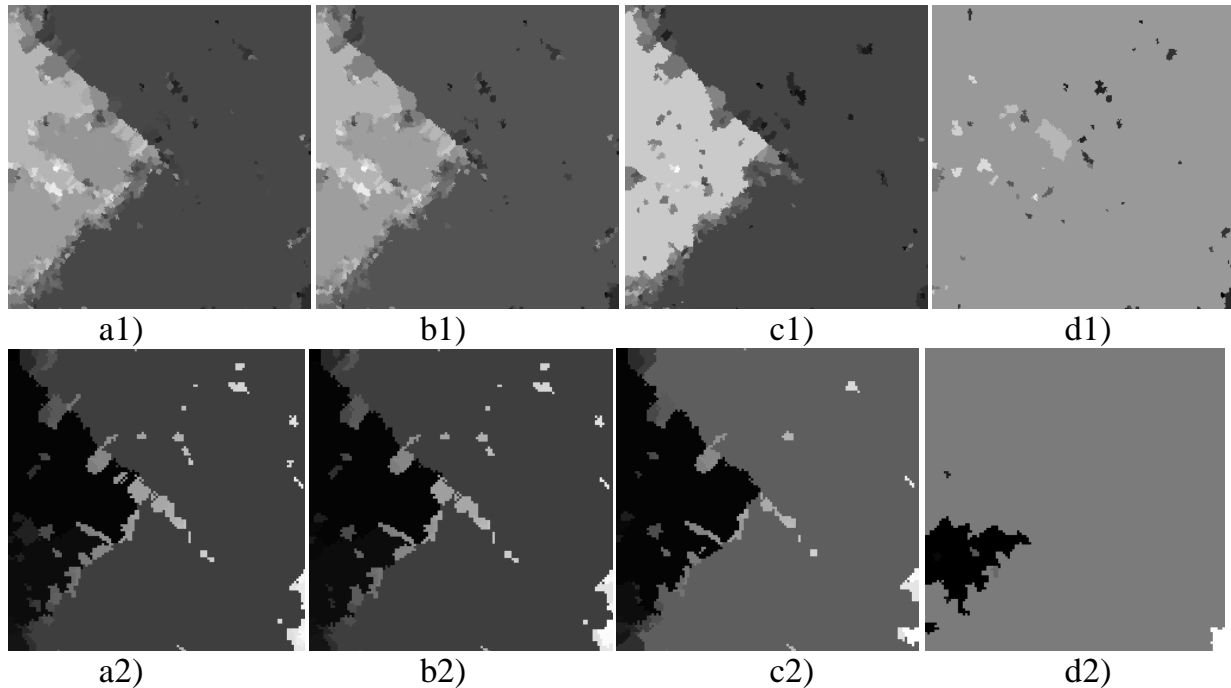


Fig. IV.12 : Résultats de la segmentation de l'image Synthétique en utilisant différents seuils de fusion

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) S=0 b) S=5 c) S=10 d) S=25

La variation du seuil de fusion peut faire varier les résultats considérablement. Lorsque sa valeur est importante, le résultat de la segmentation est mauvais tel que nous pouvons l'observer sur les figures IV.11.d et IV.12.d. Pour des valeurs assez faibles, allant de 0 à 5, les résultats apparaissent semblables et sont meilleurs. Il est bon donc de prendre une faible valeur pour S.

6. Influence du seuil de quantification QT

Lors de la fusion des classes, au cours de la quantification perceptuelle, un seuil appelé «seuil de quantification QT » est préréglé pour fusionner les classes dont les centres sont inférieurs à ce seuil. Dans ce paragraphe, nous allons tester l'influence de ce paramètre ; pour ce faire, nous fixerons les valeurs des autres paramètres et ferons varier sa valeur. Les valeurs des autres paramètres sont :

- type de l'ondelette : Haar
- niveau de décomposition de l'ondelette : 1
- seuil de fusion $S=2$
- taille de la fenêtre de balayage $w \times w=9 \times 9$.

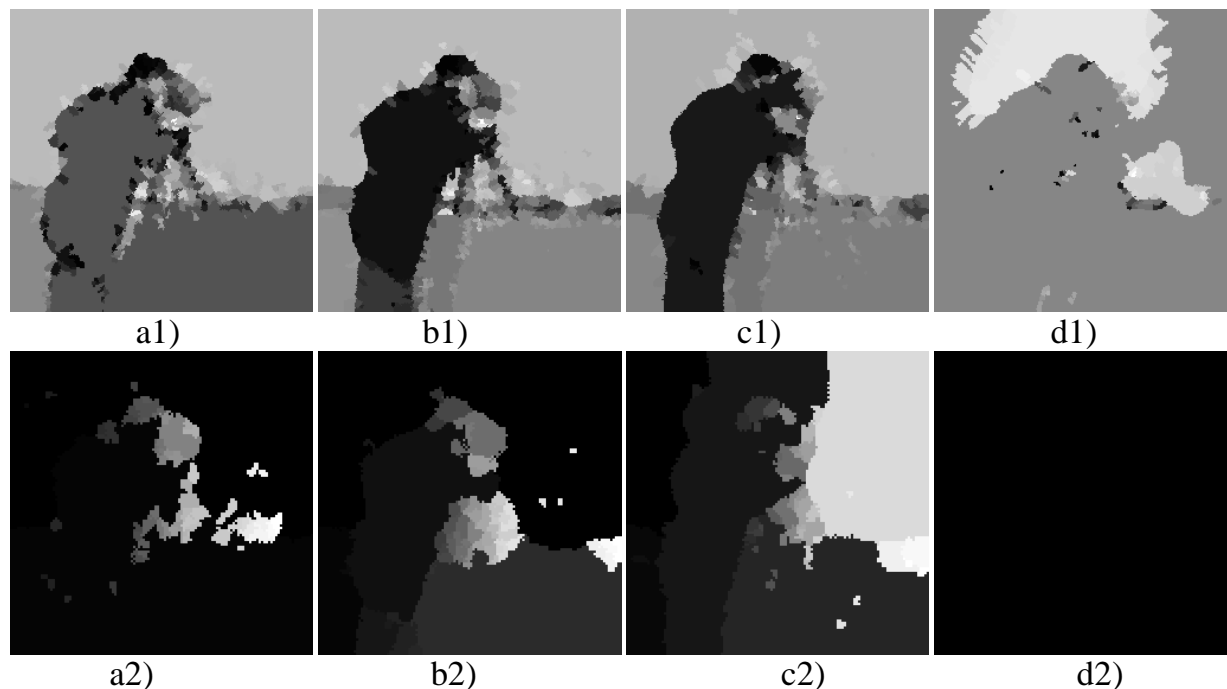


Fig. IV.13 : Résultats de la segmentation de l'image Cameraman en utilisant différents seuils de quantification

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) QT=0 b) QT=5 c) QT=25 d) QT=100

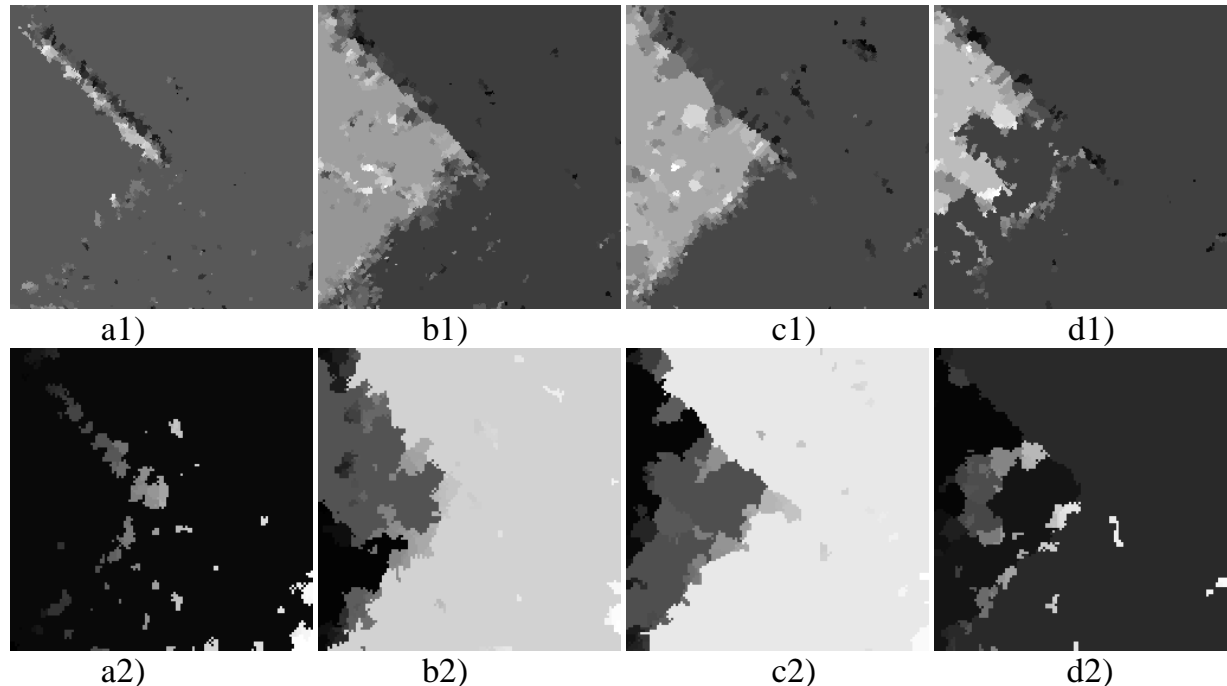


Fig. IV.14 : Résultats de la segmentation de l'image Synthétique en utilisant différents seuils de quantification

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) QT=0 b) QT=5 c) QT=25 d) QT=100

Sachant que le seuil de quantification QT est la valeur utilisée durant l'étape de quantification perceptuelle pour fusionner les classes qui sont semblables, il est donc tout à fait prévisible que lorsque ce seuil est important, les résultats seront mauvais. En effet, s'il est très important, le nombre de classe pourra être réduit à un ou deux ou à une faible valeur, ce qui ne permettra pas détecter toutes les régions de l'image. Dans la figure IV.13.d2, le nombre de régions détectés est 1 pour un seuil de quantification égale à 100. Dans les résultats présentés sur les deux figures précédentes, les résultats sont mauvais aussi dans le cas où la valeur de QT est très faible. Il faut donc faire un choix judicieux de ce paramètre. Pour la suite des tests, nous prendrons $QT=10$.

7. Influence de la taille de la fenêtre de balayage

Le calcul des poids perceptuels pour chaque pixel est réalisé en prenant en considération tous les éléments contenus dans une fenêtre centrée en ce pixel. Cette fenêtre est donc d'une taille impaire. Nous allons dans ce paragraphe étudier l'influence de cette taille sur les résultats de la segmentation. Pour ce faire, nous allons appliquer sur des images tests des fenêtres de tailles différentes tout en maintenant fixe les autres paramètres. Ces derniers auront pour valeurs :

- le type de l'ondelette : Haar
- le niveau de décomposition de l'ondelette : 1
- le seuil de quantification $QT= 10$
- le seuil de fusion $S=2$

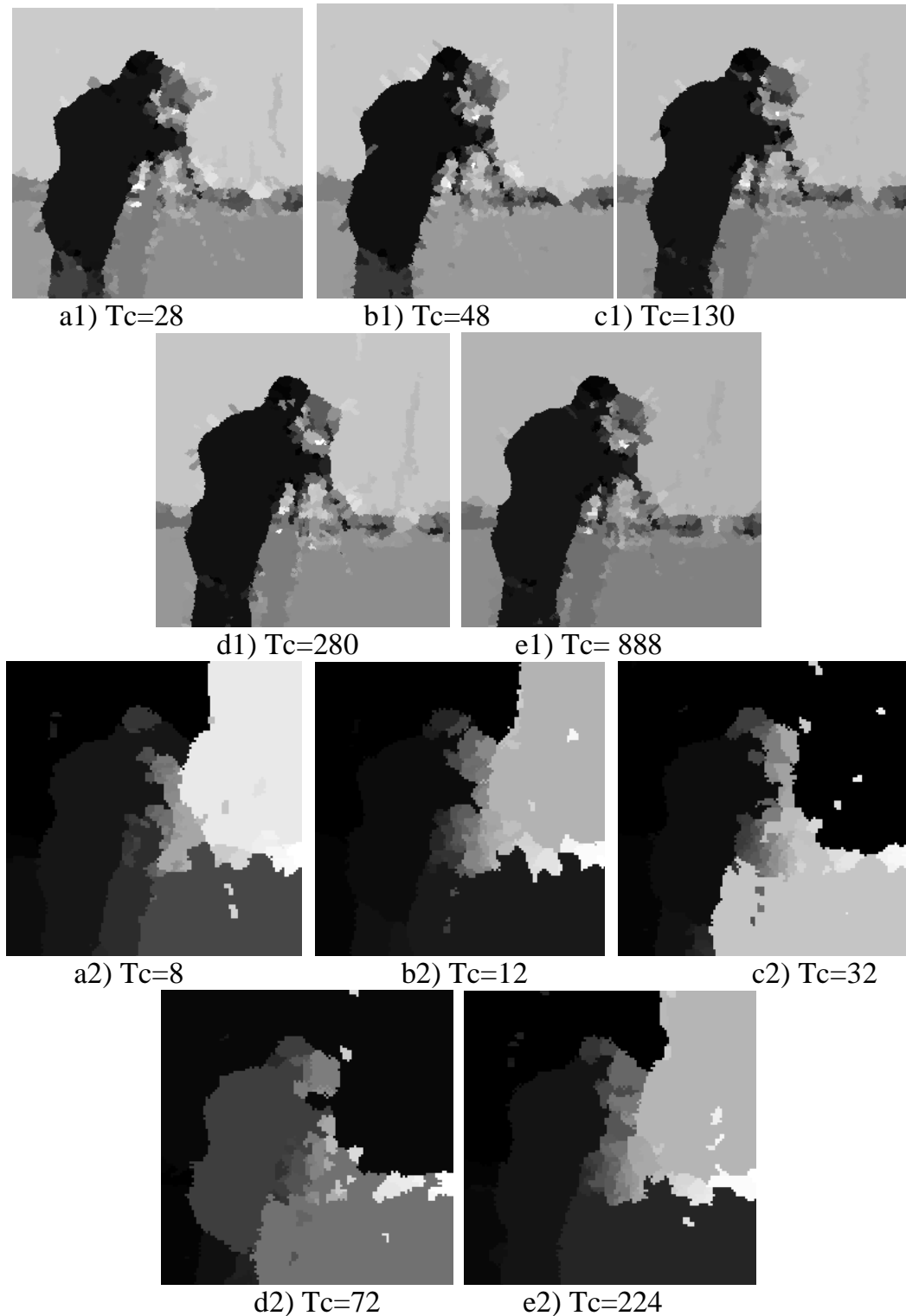


Fig. IV.14 : Résultats de la segmentation de l'image Cameraman en utilisant différentes tailles de la fenêtre de balayage

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) $w_x w_y=3 \times 3$ b) $w_x w_y=5 \times 5$ c) $w_x w_y=9 \times 9$ d) $w_x w_y=13 \times 13$ e) $w_x w_y=21 \times 21$

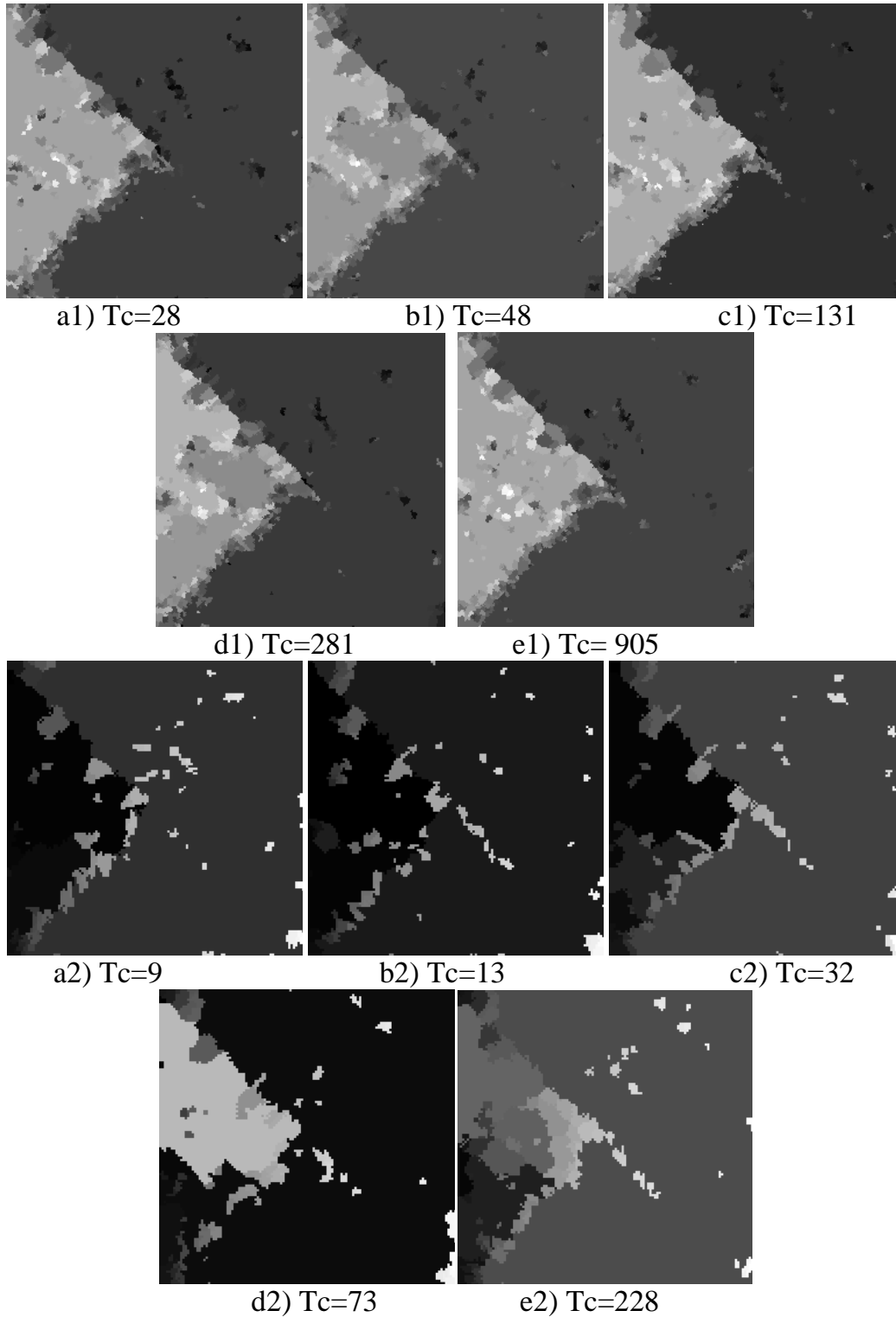


Fig. IV.14 : Résultats de la segmentation de l'image Synthétique en utilisant différentes tailles de la fenêtre de balayage

1) Résultat sans décomposition 2) Résultats avec décomposition par l'ondelette Haar au niveau 1
 a) wxw=3x3 b) wxw=5x5 c) wxw=9x9 d) wxw=13x13 e) wxw=21x21

Nous constatons que lorsque la taille de la fenêtre de balayage croît, le temps d'exécution T_c augmente considérablement. Ainsi pour une taille de $w \times w = 3 \times 3$, ce temps est d'environ 28 secondes alors qu'il est d'environ 900 secondes pour la taille $w \times w = 21 \times 21$. Sur les deux figures précédentes, nous ne constatons pas de variations significatives lorsqu'on varie la taille de la fenêtre de balayage.

8. Conclusion

Dans ce chapitre, nous avons présenté l'approche adoptée ainsi que les résultats obtenus sur une image réelle et une image synthétique. Plusieurs tests ont été réalisés afin de voir l'influence des paramètres ajustables sur la qualité des résultats.

De manière générale, la réduction de l'image par la transformée en ondelettes apporte un gain important en temps de calcul sans dégrader de manière significative la segmentation.

Conclusion générale

Nous avons présenté dans ce travail une méthode de segmentation d'images nommée JSEG. Cette méthode présente deux étapes principales qui sont la quantification couleur (ou niveau de gris) et la segmentation spatiale. Dans la première étape, une quantification couleurs (ou niveau de gris) est réalisée et une image étiquette nommée classe-map en résulte. Cette image des étiquettes subit une segmentation spatiale composée de plusieurs étapes dont la principale est le calcul des J-images.

La segmentation par la méthode JSEG nécessite un temps de calcul important vu le nombre de techniques mises en jeu. Dans ce travail, nous avons proposé de réduire ce temps de calcul en réduisant la taille de l'image par la transformée en ondelettes.

Nous avons par la suite réalisé de nombreux tests pour vérifier si cette réduction ne dégrade pas de manière significative la qualité des résultats de la segmentation en les comparant à ceux obtenus sans réduction.

La réduction de l'image par la transformée en ondelettes au premier niveau a permis de diminuer considérablement le temps de calcul sans altération nette des résultats.

Il serait intéressant de vérifier les résultats avec d'autres types d'ondelettes qui ne sont pas étudiés dans ce mémoire tel que les ondelettes biorthogonales, les symlets, etc.

La majeure partie du temps d'exécution de la technique JSEG est due à l'étape de quantification basée sur les deux algorithmes « Peer Group Filtering » et « Generalized Lloyd Algorithm ». Pour réduire ce temps, il faudra, sans doute, essayer d'autres techniques plus rapides pour cette étape.

Annexe

Les ondelettes les plus répondues sont les suivantes :

1) Ondelette de Haar

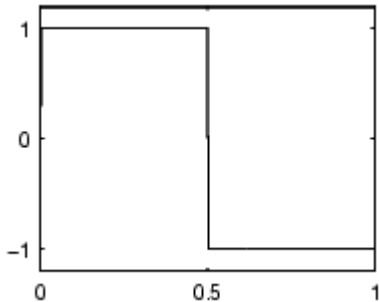


Fig.1: Allure de l'ondelette de Haar

C'est l'ondelette la plus ancienne et la plus simple. Elle est définie comme suit :

$$\Psi(t) = \begin{cases} 1 & \text{si } t \in [0, 0.5] \\ -1 & \text{si } t \in [0.5, 1] \\ 0 & \text{ailleurs} \end{cases}$$

2) Ondelette de Daubechies

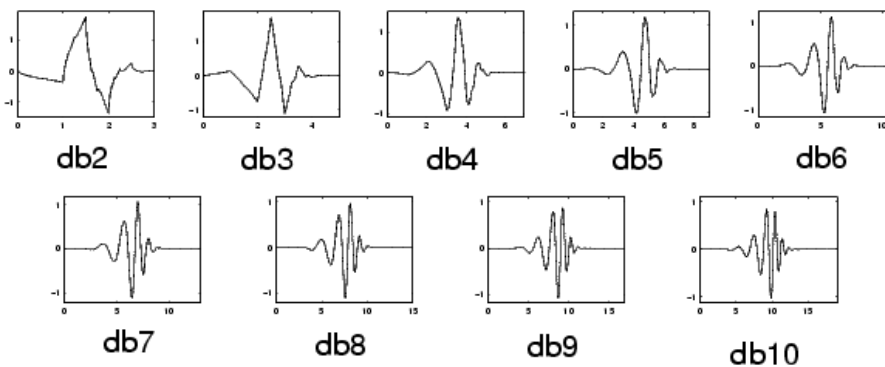


Fig.2: Allure des ondelettes de Daubechies.

Elles sont notée dbN où le N est l'ordre de l'ondelette. $db1$ est exactement la même que celle de Haar et ci-dessus sont représentées les 9 ondelettes qui suivent. Ce type d'ondelettes est très utilisé en traitement d'images.

3) Les ondelettes biorthogonales

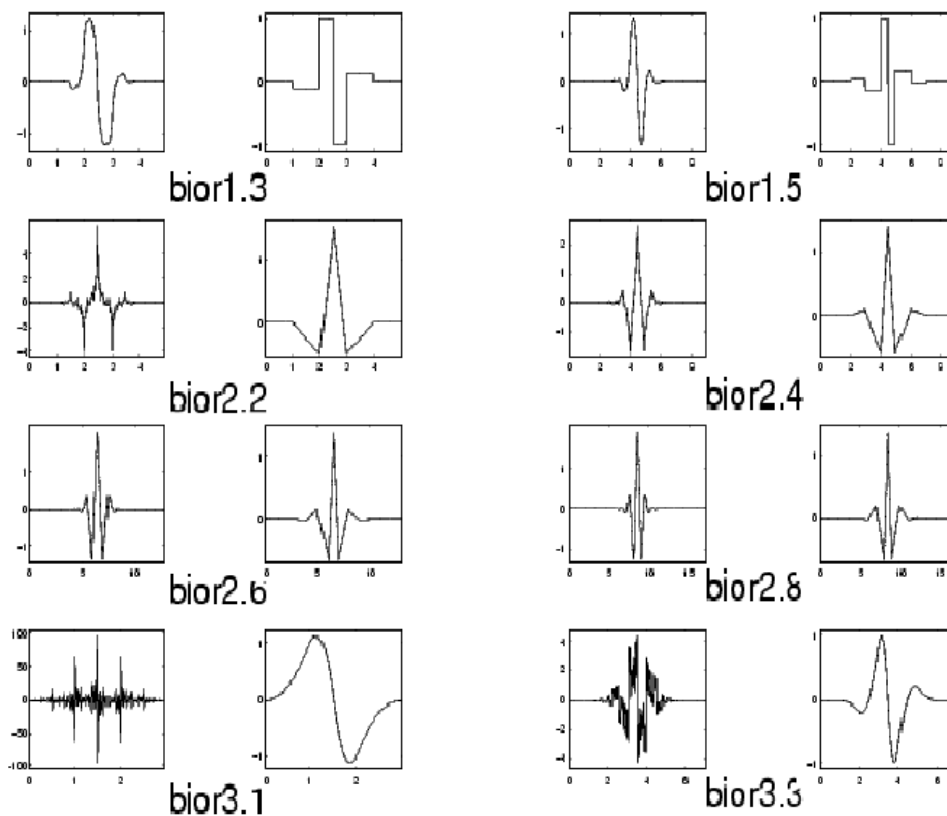


Fig.3: Allure des ondelettes biorthogonales

Ce type d'ondelettes présente la propriété de la phase linéaire qui est nécessaire à la reconstruction du signal image. En employant deux ondelettes, une pour la décomposition et l'autre pour la reconstruction.

4) Les ondelettes de coifman

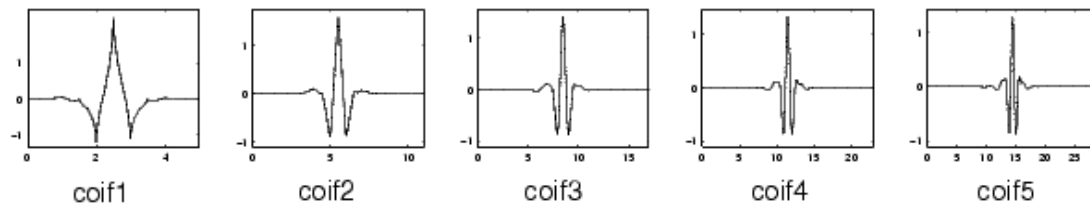


Fig.4: Allure des ondelettes de coifman

Elles sont nommées « ondelettes de coifman » et notée « coifN ».

5) Les symlets

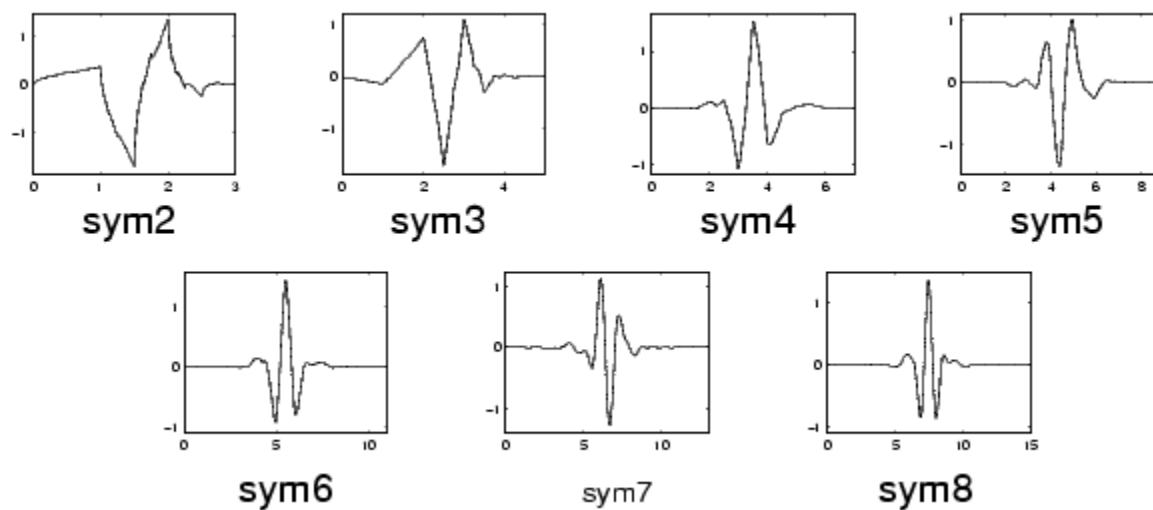


Fig.5: Allure des Ondelettes symlets

Les symlets sont des ondelettes presque symétriques, proposées par Daubechies comme modification de la famille dbN.

La propriété principale de cette famille est bien la symétrie.

6) Chapeau mexicain

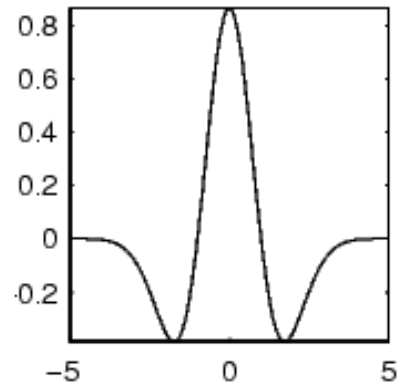


Fig.6: Allure de l'ondelette Chapeau mexicain

Elle s'écrit sous la forme suivante :

$$\Psi(t) = 2/\sqrt{3} \frac{\pi}{4} \cdot \exp(-t^2/2)$$

7) L'ondelette de Meyer

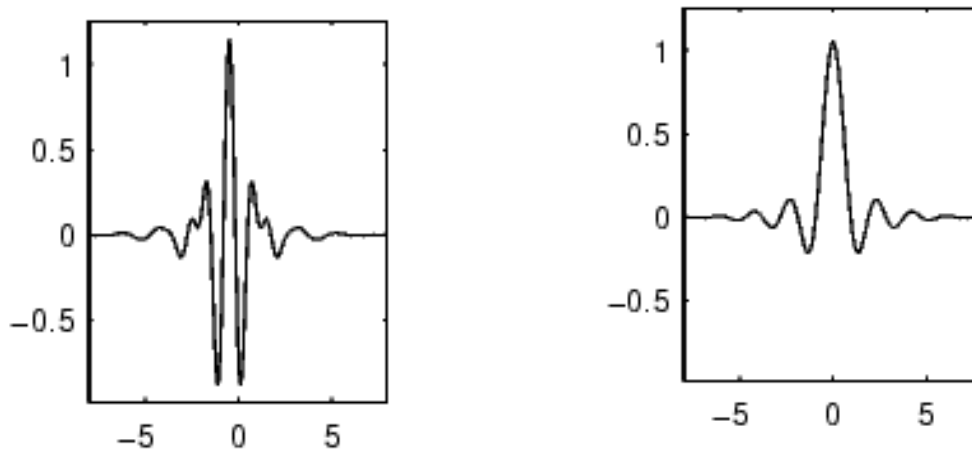


Fig.7: Allure de l'ondelette de Meyer et la fonction d'échelle

8) Ondelette de Morlet

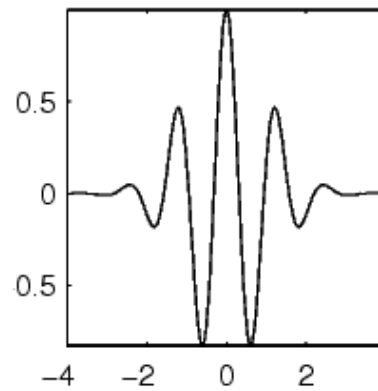


Fig.8 : Ondelette de Morlet

L'ondelette de Morlet s'écrit sous la forme suivante :

$$\Psi(t) = C * \exp(-t^2/2) * \cos(5t)$$

C : est un paramètre de normalisation en vue de la pour la reconstruction.

Notations

- \mathbb{Z} est l'ensemble des nombres entiers et \mathbb{R} est l'ensemble des nombres réels.

- $L^2(\mathbb{R})$: espace vectoriel dans \mathbb{R} des fonctions continues à une dimension à carrée intégrable.

-Le produit scalaire de deux fonctions f et g appartenant à $L^2(\mathbb{R})$ est défini par :

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} g(x)f(x)dx$$

-Norme d'une fonction définie dans $L^2(\mathbb{R})$:

$$\|f(x)\| = \sqrt{\int_{-\infty}^{+\infty} |f(x)|^2 dx}$$

-Définition de la convolution dans $L^2(\mathbb{R})$:

-La convolution de deux fonctions f et $g \in L^2(\mathbb{R})$ est définie par :

$$f * g(u) = (f(x) * g(u))(u) = \int_{-\infty}^{+\infty} f(x)g(u - x)dx$$

-La transformée de Fourier d'une fonction $f \in L^2(\mathbb{R})$ est définie comme suit :

$$F(w) = \int_{-\infty}^{+\infty} f(x)e^{-jwx}$$

- $L^2(\mathbb{R}^2)$: Espace vectoriel des fonctions continues à deux dimensions à carrée intégrable.

-Le produit scalaire des fonctions f et g dans $L^2(\mathbb{R}^2)$ est défini par :

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y) f(x, y) dx dy$$

Bibliographie

- [1] M. Bouazza, Application de la transformée en ondelettes pour la reconnaissance des chiffres manuscrits, Thèse d'ingénieur d'état, Département d'automatique, UMMTO, 2008.
- [2] G. Burel, Introduction au traitement d'images, Paris, Hemès Science publication, 2001.
- [3] Y.Deng, C.Kenney, M.S.Moore et B.S.Manjunath, Peer group filtering and perceptual color image quantization, Proc.IEEE Int'l Symp.Circuit and Systems, vol.4,1999, pp.21 - 24.
- [4] Y. Deng et B.S. Manjunath, Unsupervised segmentation of color texture regions in images and video, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, NO.8 2001, pp.800-810.
- [5] Z. Drici, T. Selmani et M. Yahyatene, Application des ondelettes biorthogonales pour le codage des images numériques, Thèse d'ingénieur d'état, Département d'électronique, UMMTO, 2002.
- [6] D. Hamel et N. Meslem, Traitement d'images sous MATLAB par la transformée en ondelettes, Thèse d'ingénieur d'état, Département d'automatique, UMMTO, 2002.
- [7] V.S. Jayanthi, K. Swapnil Marothi, T.M. Ishaq Mohammed Abbas et A.Shanmugam, Performance analysis of vector quantizer using Modified Generalized Lloyd algorithm, International Journal of Imaging Science and Engineering, vol 1, NO.1, 2007, pp.11-15.
- [8] H. Medrouk et S. Toucherift, Segmentation par classification d'une partie de l'image, Thèse d'ingénieur d'état, Département d'automatique, UMMTO, 2010.
- [9] H. Messaoudi, N.Ould-Dris & S.Zehraoui, segmentation d'image par la transformée en Ondelette sous MATLAB, Thèse d'ingénieur, Département d'électronique, UMMTO, 2003.
- [10] F. Truchtet. « Ondelettes pour le signal numérique ». Paris, Hemès, 1998.
- [11] <http://www.wavelet.org>
- [12] <http://www.ondelette.com>
- [13] <http://www.amara.com/current/wavelet.html>
- [14] <http://stat.stanford.edu/~wavelab>

[15] <http://www.multires.caltech.edu/>

[16] <http://www.cs.nyu.edu/cs/faculty/mallat/>

[17] <http://faculty.gvsu.edu/aboufadi/web/dw.htm>