République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la recherche scientifique

Université Dr YAHIA FARES de MEDEA



Faculté de sciences et de la technologie

Département de génie Electrique et de l'informatique

Mémoire de fin d'études pour l'obtention du diplôme de Master en Génie électrique

Options : **Systèmes de Télécommunications.**

Thème :

Etude de la technique du filtrage paramétrique en bloc en exploitant les bases d'ondelettes pour la segmentation des images fixes

Proposé et dirigé par :

Mr : MIMEN Bekhti

Etudié par :

Mr : GHEZALI Abdelhamid

Mlle : MADAT Amina

Promotion 2015/2016

Dédicace

Je dédie ce modeste travail à :

Ma très chère mère, et mon très cher père,

à leur grand sacrifice et le dévouement pour mon

bonheur.

Mes frères et ma soeur,et toute la famille sans exception.

Toute mes amis et toute personne qui mon aidée de loin ou

de prés.

AMINA

Dédicace

Je dédie ce modeste travail à tous ceux qui me sont chers :

A ma très chère mère et mon père, pour son immense amour

pour sa patience dont ils ont fait preuve à mon égard,

soutien moral, ses conseils précieux et les sacrifices

qu'il son toujours consentit.

A mon cher frère et sœur, À tous les membres de ma grande famille.

A tous mes amis, pour leurs encouragements

et leurs aides considérables.

Que ce mémoire soit l'expression de mon profond

respect et mon grand amour.

ABD ELHAMID

Remerciement

Nous rendons grâce à Dieu qui nous a donné l'aide, la patience et le courage pour accomplir ce travail.

Je tiens à adresser mes plus vifs remerciements aux:

Notre promoteur Mr MIMEN Bakhti, qui nous a aidé et donné tout son temps pour compléter notre travail. Nous lui remercions pour sa compréhension, ses conseils précieux, sa bonne orientation et sa présence permanente au près de nous.

Nos vives reconnaissances vont à tous les enseignants qui ont collaboré à notre formation au sein du tronc commun et l'institut d'électronique de l'université Yahia Farès de MÉDÉA.

Nos remerciements vont aux au membre de juré qui ont accepté de juger notre travail.

Tous nos remerciements également à nos familles pour l'aideinconditionnel durant cette période.

Enfin, Nous remercions sincèrement tous nos amis de pris ou de loin, et tous ceux qui nous a aidé, d'une manière ou d'une autre durant toute la période de recherche de ce mémoire

ملخص

يهدف هذا المشروع إلى دراسة وعرض طريقة الكشف عن حواف الصورة الثابتة، اعتمادا على تقنية تسمى ***الترشيح الوسيطي**,* والتي تعتمد على تحليل الصورة أفقيا و عموديا بمعايير أساسية لهذا الكشف.

من جهة أخرى اقترحنا طريقة لتحسين التقنية السابقة و ذلك بتقسيم الصورة إلى أجزاء صغيرة الحجم وان تطبيق مستوى العتبة هو ضروري لتحسين أداء الكشف عن الحواف.وعلاوة على ذلك،فإن استخدام مختلف النوافذ يسمح بتحسين الكشف عن حواف الصورة حيث في النهاية جرت دراسة و مناقشة النتائج المتحصل عليها.

كلمات مفتاحية : تحليل صورة،تحديد الحواف،تجزئة، الترشيح الوسيطي،حساب التداخل ,شريحة،مويجة ، dll2,dkl

Résumé

Le but en ce projet est consacré à l'étude et la simulation de la détection de contours des images fixes par la méthode du filtrage paramétrique suivant une analyse par ligne et par colonne en exploitant les paramètres nécessaire à cette détection.

D'autre part nous avons proposé une approche d'amélioration en basant sur la décomposition en sous blocs de petites tailles. Une mise en œuvre a été effectuée suivant les divers paramètres de détection et de seuillage ainsi que des types de fenêtres choisis ; à la fin une discussion des résultats obtenus a été faite.

Mots clés : Traitement d'image, détection de contour, segmentation, filtrage paramétrique, mesure de distorsion, autocorrélation, ondelette, DKL, DLL2.

Abstract

The goal in this project is dedicated to the study and simulation of edge detection still images by the parametric filtering method by following a line analysis by column operator settings required for this detection.

Furthermore we have proposed an improvement approach based on the decomposition into sub-blocks of smaller sizes. An implementation was performed according to the various detection parameters and thresholding and selected type of windows; at the end a discussion of the results was made.

Keywords: Image processing, edge detection, segmentation, parametric filtering, distortion measurement, correlation, wavelet, DKL, DLL2.

INTRODUCTION GÉNÉRALE

Introduction générale

La technologie numérique moderne est devenue omniprésente. Grâce à elle, il est devenu possible de traiter des signaux multidimensionnels avec des systèmes très divers, depuis les téléphones portables jusqu'aux ordinateurs massivement parallèles. Dans le domaine des images

Le traitement d'images devient un but très intéressant au niveau de l'industrie de la médecine et au niveau militaire et plusieurs domaines ou le traitement d'image base sur certaines fonctionnalités telles que l'analyse et la synthèse par exemple, l'exploitation des informations partielles pour restaurer l'information totale.

Parmi les traitements des signaux existants, on trouve la segmentation des images par région ou par contour, en exploitant divers techniques, telles que celles de CANNY et DERRICHE.

Dans ce mémoire nous allons entamer, d'une part une étude de la technique de segmentation des images fixes en contours par la méthode du filtrage paramétrique sous divers paramètres, et d'autre part nous allons proposer une amélioration de cette technique PF par traitement en blocs de petites tailles suivants divers paramètres tels que le choix du type de la fenêtre du filtrage, la taille de bloc et des segments, ainsi que le type et les niveaux de seuillage

Notre mémoire est organisé comme suite :

- Le premier chapitre présente des généralités sur les images et leurs méthodes de traitement (définition, typologie et caractéristiques, les différentes fenêtres et ondelettes).
- Au deuxième chapitre nous allons étudier quelques techniques de détection de contours.

- Le troisième chapitre est consacré à la présentation de la méthode de filtrage paramétrique.
- Dans le dernier chapitre on présente les résultats obtenus par l'utilisation de notre application.

CHAPITRE I

Généralités sur le traitement d'images

Chapitre I :

Généralités sur le traitement d'image

I.1. Introduction

L'image est l'un des moyens les plus importants qu'utilise l'homme pour communiquer. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinente.

I.2. Notions générales sur l'image

I.2.1. Définition

Une image est un support d'informations présentant les éléments d'une scène qui a été capturé par un appareil. **[1]**

En traitement de signal, une image est définit comme étant un signal bidimensionnel.

En mathématique, une image est une application d'un sous-ensemble MxN de RxR vers l'ensemble des entiers Z qui a chaque couple de réels (x,y) associe l' entier $f(x,y)$.

$f : M\text{x}N \longrightarrow Z$

$(x,y) \longrightarrow f(x,y)$

L'image peut donc être décrite par une fonction $f(x,y)$ tel que :

- f : la fonction d'intensité lumineuse définie sur un domaine borné;
- x,y : les coordonnées linéaires d'un point de l'image.

Cependant, sous cette forme, l'image reste inexploitable dans le domaine informatique, d'où la nécessité de la numérisation.

I.2.2. La taille d'une image

Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image**. [2]**

I .2 .3 Définition d'image numérique

Une image numérisée est un ensemble de pixels (Picture cell), à chacun desquels est attribué un niveau de gris, de 0 (noir) à 255 (blanc). Le nombre $256 = 2^8$ traduit le codage sur 8 bits en binaire, p.ex. 10010101.

Une image numérisée en niveaux de gris n'est donc rien d'autre qu'une fonction définie sur une grille discrète et prenant des valeurs discrètes, c'est à dire un tableau (matrice) de nombres (binaires). **[3]**

Une telle image contient donc une quantité d'information énorme. Toutefois, de par sa définition même, elle va présenter des difficultés géométriques : courbes et droites sont mal reproduites, il y a des effets de pixellisation

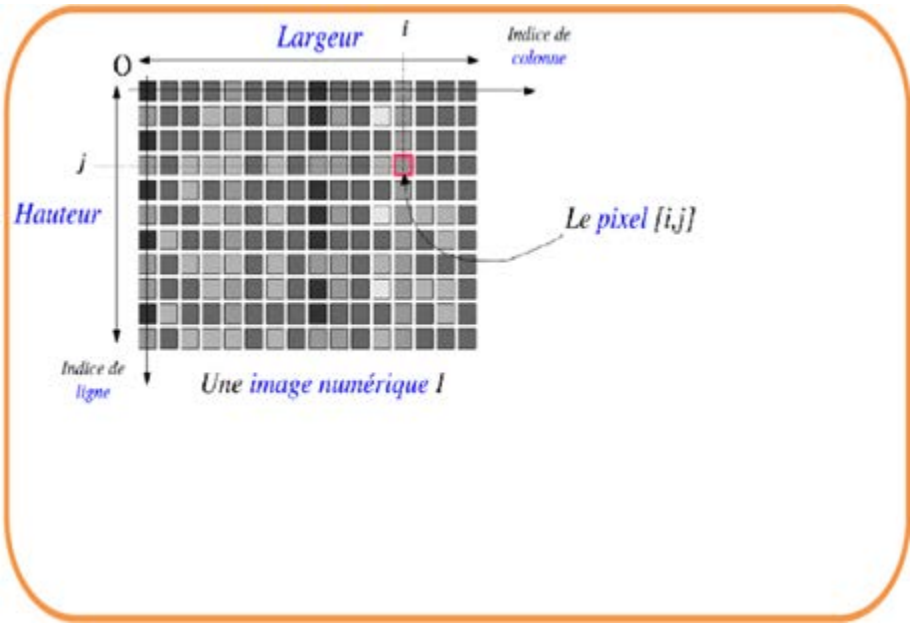


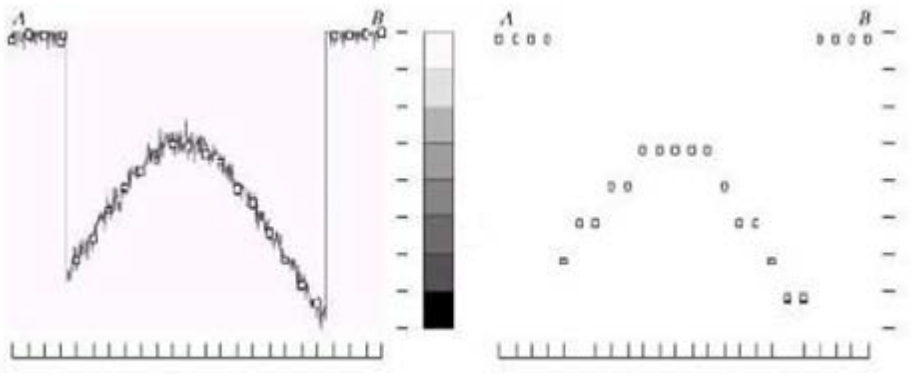
Figure I.1.*Représentation d'une image numérique*

I.2.4 Caractéristiques d'une image numérique

a. Échantillonnage et quantification

L'échantillonnage est limité par la capacité du capteur, donc le nombre de pixels disponible (ou autre limite imposée). **[4]**

La quantification est limitée par la quantité de tons (de gris) définie dans l'intervalle.



Echantillonnage Quantification

Figure I.2.*Echantillonnage et quantification.*

I.2.5. Pixel

On appelle «pixel» la plus petite zone adressable d'une image sur écran, soit le plus petit élément que l'on puisse activer sur l'écran.

Exemple: La lettre A, peut être affichée comme un groupe de pixels dans la figure ci-dessous:

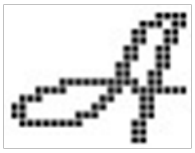


Figure I.3. *Affichage d'un groupe de pixels*

a. Distance entre 2 pixels

L'image est caractérisé par un couple de coordonnée (X, Y).

Pour deux pixels P (Xp, Yp) et Q (Xq, Yq) :**[4]**

- Distance de ***manhattan***(city block):

$$D(P, Q)=|\mathbf{X_p} - \mathbf{X_q}| + |\mathbf{X_p} - \mathbf{X_q}| \qquad (I.1)$$

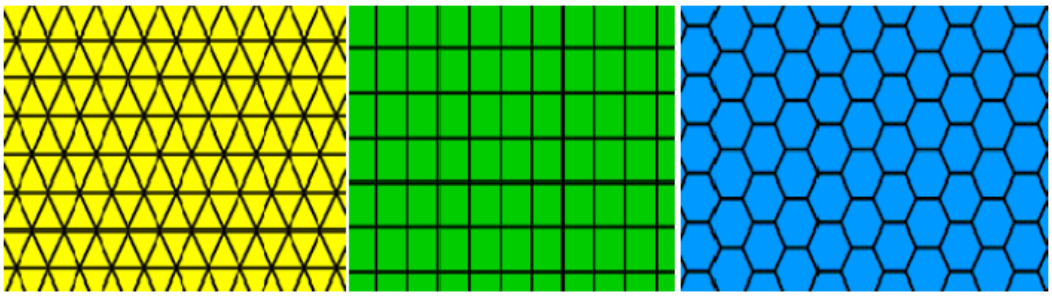
- Distance de l'***échiquier*** :

$$D(P, Q) = \text{Max}\left(|X_p - X_q|, |Y_p - Y_q|\right) \quad \text{(I.2)}$$

b.Pavages du plan :

Un pavage du plan est une partition du plan en cellules élémentaires (pixels).

Il n'existe que 3 pavages réguliers du plan :



(a) (b) (c)

Figure I.4. *(a) pavage Triangulaire, (b) pavage carré, (c) pavage hexagonal*

I.2.6 Voisinage, maillage et connexité d'un pixel

a) Voisinage

Un pixel représenté par son intensité ou niveau de gris M(x, y), a quatre voisins aux points: **M5 (x+1, y)**, **M4 (x-1, y)**, **M7 (x, y+1)**, **M2 (x, y-1)**. Cet ensemble de pixels appelé quatre voisins de M (x, y) représente les pixels aux quatre points cardinaux de M (x, y).

Les voisins diagonaux de M (x, y) se trouvent aux points: **M8 (x+1, y+1)**, **M3 (x+1, y-1)**, **M6 (x-1, y+1)**, **M1 (x-1, y-1)**. L'ensemble des points cardinaux et diagonaux de M (x, y) est appelé huit voisins de M (x, y) appelés aussi 8-connexité. **[1]**

0 X

M1(x-1,y-1)	M2(x,y-1)	M3(x+1,y-1)
M4(x-1,y)	M(x,y)	M5 (x+1,y)
M6 (x-1,y+1)	M7 (x,y+1)	M8 (x+1,y+1)

Y

TableauI.1: *Voisinage d'un pixel.*

b) Maillage

Le maillage est l'arrangement géométrique des pixels dans l'image. On utilise trois types qui résultent de différents dessellassions du plan par des figures géométrique:

- Maillage carré: qui correspond à la réalité physique des capteurs d'image.
- Maillage hexagonal: est une référence pour le domaine de la morphologie mathématique.
- Maillage triangulaire : est plus confidentiel.

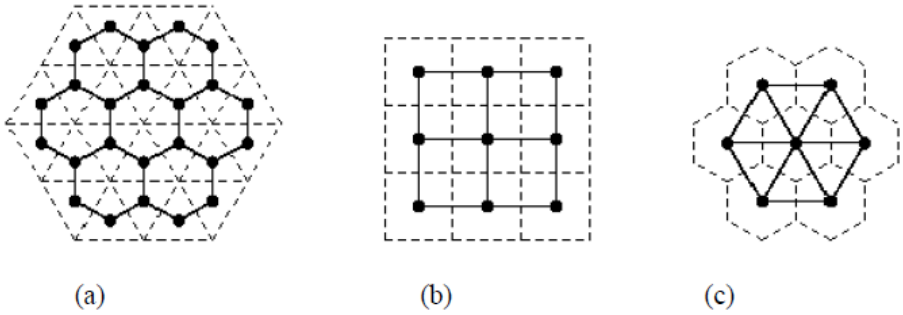


Figure I.5. *(a)Maillage hexagonal, (b) Maillage carré, (c) Maillage triangulaire*

c) Connexité

Deux points M et P sont connexes s'ils sont mutuellement voisins par le système de voisinage défini. **[2]**

Il y a 3 grands types classiques de connexité

- 4 connexités (Maillage carrée)
- 6 connexités (Maillage hexagonale)
- 8 connexités (Maillage triangulaire)

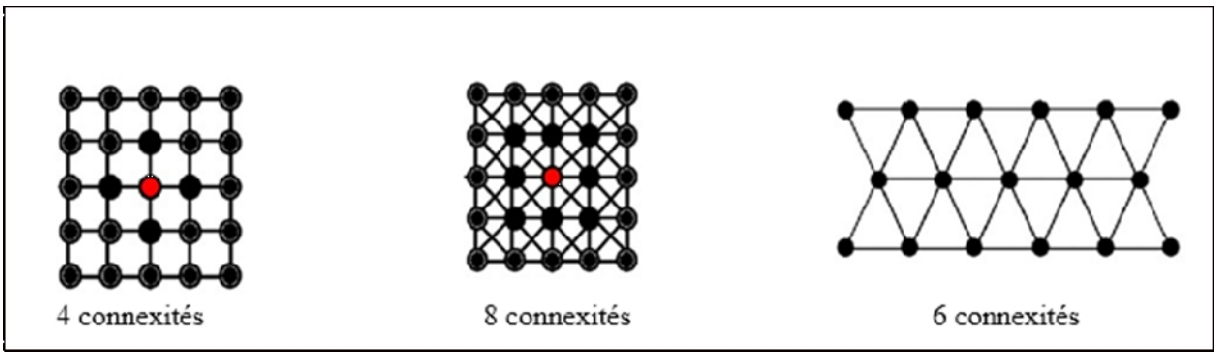


Figure I.6.*Connexité des pixels*

I.2.7. Résolution

La résolution est définie par un nombre de pixels par unité de longueur de l'image à numériser en dpi (dots per inch) ou ppp (points par pouce). On parle de définition pour un écran et de résolution pour une image .Plus le nombre de pixels est élevé par unité de longueur de l'image à numériser, plus la quantité d'information qui décrit l'image est importante et plus la résolution est élevée (et plus le poids de l'image est élevé).La résolution d'une image correspond au niveau de détail qui va être représenté sur cette image. **[5]**

I.2.8. Histogramme

L'histogramme est la représentation graphique de la proportion de pixels d'une image par gamme de luminosité, l'histogramme permet de répondre concrètement aux questions suivantes : « Combien de pixels de mon image sont-ils dans les tons foncés (ou la haute lumière) ? », ou encore « Quelle est la luminosité majoritaire dans mon image? ». Du fait qu'il s'agit d'une représentation graphique, l'histogramme permet de répondre à ces questions sans calcul complexe. **[6]**

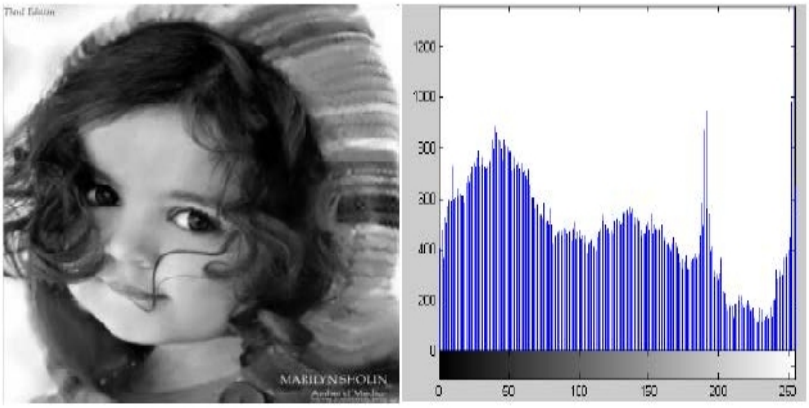


Figure I.7. *L'histogramme d'une image noir et blanc.*

a) Utilisation de l'histogramme

L'histogramme consiste à pouvoir isoler des objets du fond dans le cas d'une image à fort contraste comme l'illustre la figure I.8 et la figure I.9 le premier pic correspond aux objets plus foncés, le second au fond plus clair.

L'histogramme est très utile pour choisir le seuil qui permettra de définir «le mieux possible »le contour de l'objet.

L'histogramme permet la correction des niveaux de gris, en améliorant le contraste et en renforçant la dynamique de l'image et ceci grâce à l'égalisation de l'histogramme**.**

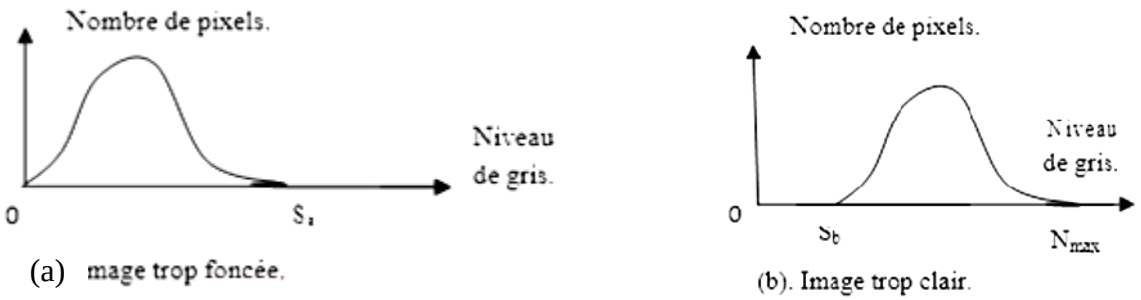


Figure I.8. *Histogramme d'image*

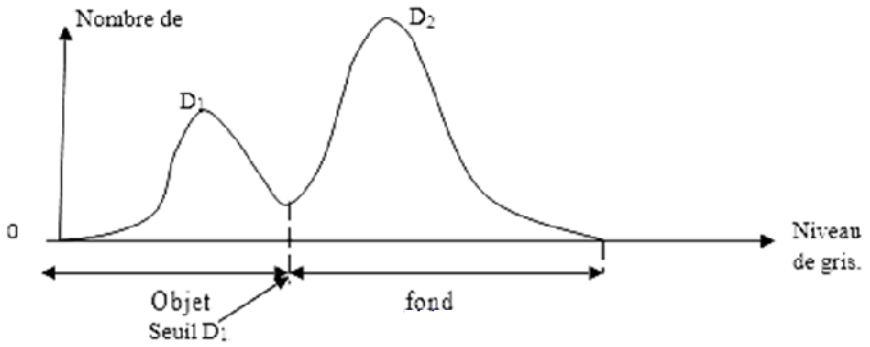


Figure I.9.*Utilisation de l'histogramme pour séparer des objets du fond.*

- ***Qualité de numérisation***

  L'histogramme discret permet de juger si l'opération du numérisation est correcte regard de l'adaptation de la dynamique de densités (D_{min},D_{max}) à la dynamique des niveaux du quantificateur (D_0,…,D_k,…,D_{n-1}).

- ***Choix du seuil***

  Le choix du seuil dépendant de l'histogramme, permet de mieux définir le contour de l'objet. Celui-ci présente deux pics, l'un lié aux fonds, l'autre à l'objet. Il indique un point minimum où le seuil s'intègre, car il s'agit du point où la fonction de surface S(d), varie le plus lentement, ce qui est moins sensible pour le choix de D.

- ***Egalisation***

  La transformation d'égalisation est construite de telle façon que l'histogramme de l'image transformée soit le plus plat possible. Cette technique améliore le contraste et permet d'augmenter artificiellement la clarté d'une image grâce à une meilleure répartition des intensités relatives**,** L'égalisation de l'histogramme d'une image est une méthode qui vise à assurer une distribution homogène des valeurs dans la totalité de la plage dynamique des valeurs possibles.

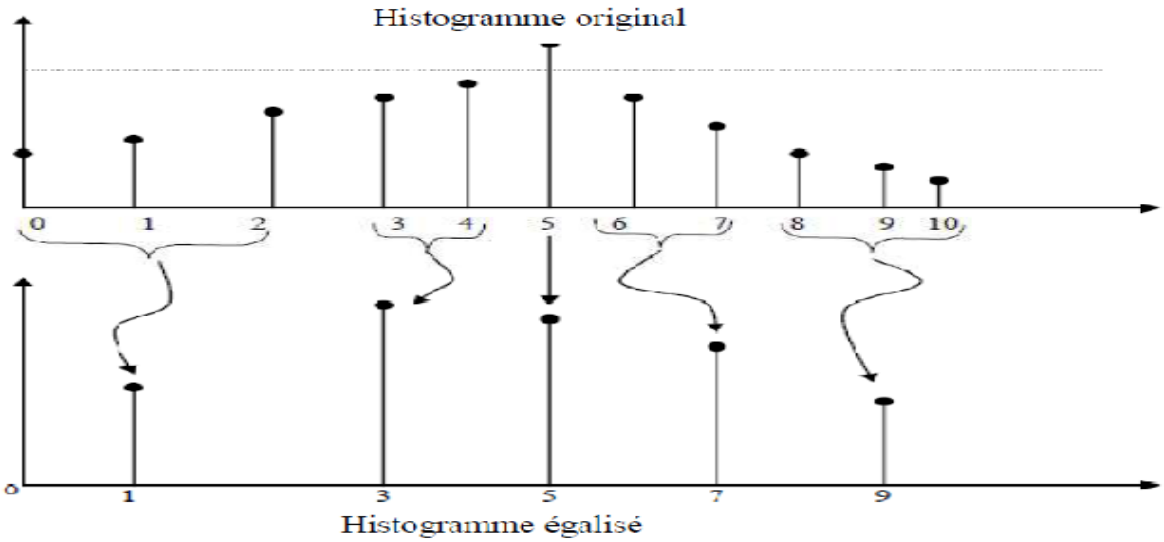


Figure I.10. *Principe d'égalisation d'histogramme*

I.2.9Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant la mesure photométrique de l'intensité du rayonnement d'une source lumineuse, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par. **[7]**

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

I.2.10 Contraste

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

$$\boldsymbol{C = \frac{(L1-L2)}{(L1+L2)}} \qquad \text{(I.3)}$$

I.3. Segmentation

La segmentation consiste à créer des partitions dans l'image en identifiant des pixels similaires. Le critère de similarité permet de contrôler l'aspect final de la segmentation (regroupement par couleur, par homogénéité, par taille,…).

La segmentation a pour objectif de différencier des zones d'intérêt (par exemple objets/fond).C'est généralement une première étape d'un traitement plus complexe comme le filtrage adaptatif ou la reconnaissance de forme. **[8]**

Un algorithme de segmentation s'appuie donc sur les étapes suivantes:

- La recherche d'homogénéité locale pour définir les régions.
- La recherche de discontinuité sa fin démettre en évidence les contours.
- Ou encore sur la coopération des deux principes.

I. 3.1. Les différents types de segmentation

a) Approches régions

Les approches de segmentation régions visent à créer une partition de l'image en un ensemble de régions homogènes au sens d'un ou plusieurs critères. On cherche donc à rassembler les pixels par des critères de similarité.

b) Approches frontières

Contrairement aux approches régions, qui cherchent à former des zones homogènes, les approches contours travaillent sur les discontinuités des images afin de déterminer les contours des régions.

Une des opérations élémentaires en traitement d'images est la recherche des frontières ou contours. Quatre approches sont utilisées pour détecter les contours dans une image : **[9]**

- Modèle déformable (contours actifs,…)
- Méthodes différentielles (gradient, laplacian,…)
- Méthodes par ' template' (roberts, prewitt, sobel,…)
- Méthodes par optimisation basées sur modélisation des contours, de bruits, de mesure de qualité de détection, (canny,…).

b) Approches par convolution

C'est l'opérateur de base du traitement linéaire des images. Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, ce n'est que plus tard qu'on a fourni des justifications physique et fait le lien théorique avec les filtres et le traitement du signal**. [10]**

- **principe**

Soient I une image numérique et h une fonction de $[x_1,x_2]\times[y_1,y_2]$ à valeurs réelles, dite noyau de la convolution.
La convolution de I par h est définie par :

$$(I * h)[x,y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h[i,j].I[x-i,y-j] \qquad \text{(I.4)}$$

C'est-à-dire pour calculer une convolution, on remplace la valeur de chaque pixel parla valeur du produit scalaire entre les valeurs du noyau de convolution et les valeurs du voisinage du pixel considéré (par rapport à l'origine (0,0) du noyau de convolution).

- **Propriétés de la convolution**

Commutativité : $h * g = g * h$ (I.5)
Associativité :$(h * g) * k = h * (g * k) = h * g * k$ (I.6)
Distributivité : $h * (g + k) = (h * g) + (h * k)$ (I.7)

I.4. Le bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. **[6]**
Il faut noter que le bruit est non seulement lié au matériel utilisé, mais aussi à l'image elle-même : parties fortement contrastées, parties plus ou moins sombres.

I.5. Le filtrage

I.5.1. Définition

Le filtrage est le moyen qui nous permet d'atténuer au maximum le bruit et les irrégularités rapides de l'image plus rapides que les détails que l'on cherche à conserver. 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.

Il y a deux façons principales d'approcher les techniques de filtrage : **[11]**

a) la première est de s'attache aux agrégats des pixels qui composent l'image, on considère d'abord leurs attributs et parmi ceux-ci l'intensité lumineuse au point (x, y) ou ils se trouvent, les technique traitent les niveaux de gris seront donc des procédés du domaine spatial, c'est-à-dire du plan (x, y) de l'image.

b) la deuxième façon est d'opérer dans le domaine fréquentiel, le moyen d'accès est la transformée de Fourier.

I.5.2. Types des filtres

a) Filtre passe-bas

Ce filtre n'affecte pas les composants de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence.

b) Filtre passe-haut

Le filtre passe-haut a les caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composants de haut fréquence dans les données d'un signal, mais doit atténuer les composantes de basse fréquence.

c) Filtre passe-bande

Cette opération est une dérive du filtre passe-bas. Elle consiste à éliminer la redondance d'information entre l'image obtenue par filtrage passe-bas. Seule la différence entre l'image source et l'image traitée est conservée.

I.6L'autocorrélation

Un processus constitué de variables aléatoires indépendantes de même distribution est un processus de type "*bruit blanc*". On éprouve donc le caractère aléatoire des résidus au moyen d'un test de bruit blanc. La statistique de test la plus simple est l'autocorrélation des résidus, la corrélation entre la série $\{e_t\}$ et la série $\{e_{t-k}\}$ pour *k* fixé (1, 2, ...).

Il faut définir un concept de population pour formuler le test d'hypothèse. Mais la classe est trop vaste et il faut supposer la stationnarité du processus aléatoire pour pouvoir définir une autocorrélation de retard k du processus. **[12]**

Un processus $\{Y_t\}$ est stationnaire du second ordre si et seulement si les $Y_t$ont les mêmes espérances mathématiques ou moyennes $E(Y_t)$ notée *m*, et des variances identiques notéesγ_0.

En outre que l'auto-covariance $\gamma_t = E((Y_t - m)(Y_{t-k} - m))$ ne dépend pas de *t*. On peut alors effectuer un test individuel (pour un retard *k*) ou un test global.

I.7. Le seuillage

Une image numérique, pour pouvoir être exploitée a généralement besoin d'être simplifiée, c'est le but du seuillage. Ce dernier consiste à transformer l'image codée sur 6, 8 ou 16 bits, en une image binaire ou les pixels à 1correspondent aux objets et les pixels à 0 au fond de l'image. **[7]**

A la différence des différents traitements de l'image numérique, le seuillage est un passage obligatoire pour toute analyse morphologique ultérieure. Le seuillage permet de sélectionner les parties de l'image qui intéressent l'opérateur, par exemple 2 types de grains (blancs et sombres) dans un mélange. On peut donc, par exemple, attribuer à tous les pixels de l'image numérique qui ont un niveau de gris compris entre deux valeurs i1et i2, choisies par l'opérateur, la valeur 1;à tous les autres pixels est attribuée la valeur 0.

Après seuillage, les parties de l'image sélectionnées seront traduites en noir et blanc. L'image, digitalisée par l'ordinateur (0et1), est appelée image binaire. Cette dernière, tout comme l'image numérique contient des informations superfétatoires, qu'il convient d'éviter, ou masquées qu'il faut révéler. Les traitements suivants permettent de modifier l'image binaire à ces fins.

Il n'est pas en général nécessaire de s'intéresser à tous les paramètres de base relatifs à une image, car seules certaines propriétés sont utiles. L'idée est donc de transformer l'image binaire pour n'accéder qu'aux informations qui nous préoccupent. De plus, le seuillage de l'image peut faire apparaître des défauts dus à l'irrégularité de la surface ou à d'autres causes externes. La morphologie mathématique permet de transformer ces images en «tout ou rien» grâce à des éléments structurant.

I.8. Types d'images

On distingue plusieurs types d'images à la composition et au comportement différent : les images matricielles et les images vectorielles. **[13]**

I.8.1. Images matricielles (ou images bitmap)

Elle est composée comme son nom l'indique d'une matrice (tableau) de points à plusieurs dimensions, chaque dimension représentant une dimension spatiale (hauteur, largeur, profondeur), temporelle (durée) ou autre (par exemple, un niveau de résolution).

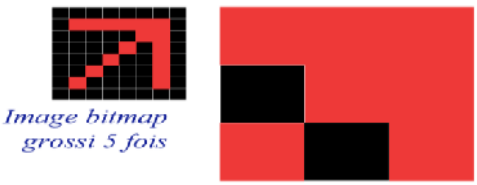


Figure I.11. *Exemple d'image matricielle*

I.8.2. Images vectorielles

Le contenu de l'image est défini par des vecteurs dont les coordonnées sont les coordonnées de points représentant une forme.

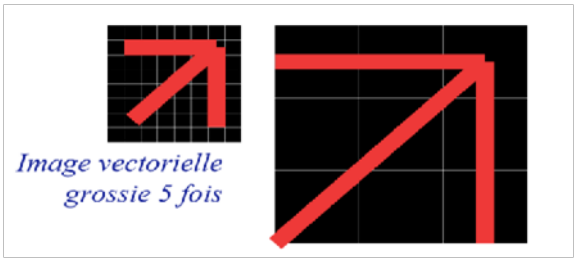


Figure I.12. *Exemple d'image vectoriel*

I.8.3. Images en couleurs

Une image couleur correspond à la synthèse additive de 3images, rouge, vert et bleu. Chaque pixel est donc codé sur 3×Nbits.

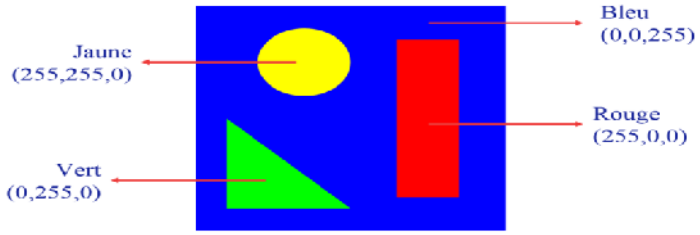


Figure I.13. *Exemple d'image en couleur*

I.8.4. Images binaire

Dans une image binaire, les pixels sont représentés par deux états logiques 0 (noir) et 1 (blanc). C'est un codage de l'image sur 1 bit.

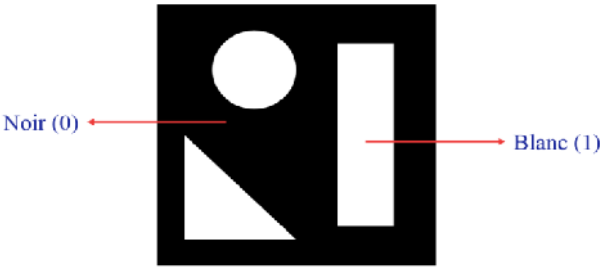


Figure I.14. *Exemple d'image binaire*

I.8.5. Images en niveau de gris

Chaque pixel est codé sur N bits, ce qui lui confère des valeurs entières comprises entre 0 (noir) et $(\mathbf{2})^{\mathbf{N}} - \mathbf{1}$ (blanc).

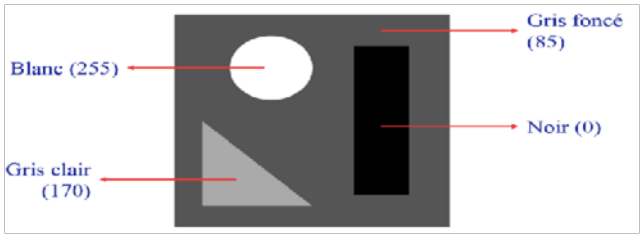


Figure I.15. *Exemple d'image en niveau de gris*

I .9. Comparaison entre les formats d'image

	Type (matriciel/ vectoriel)	Compression des données	Nombre de couleurs supportées	Affichage progressif	Animation	Transparence
JPEG	matriciel	Oui Réglable (avec pertes)	16 millions	Oui	Non	Non
JPEG2000	matriciel	Oui avec ou sans pertes	4 milliards	Oui	Oui	Oui
GIF	matriciel	Oui Sans pertes	256 maxi (palette)	Oui	Oui	Oui
PNG	matriciel	Oui Sans pertes	Palettisé (256 couleurs ou moins) ou 16 millions	Oui	Non	Oui (couche Alpha)
TIFF	matriciel	Compression ou pas avec ou sans pertes	de monochrome à 16 millions	Non	Non	Oui (couche Alpha)
BMP	matriciel	pas de compression pas de perte de qualité.	16,7 millions de couleurs	Oui	Non	Non

TableauxI.2. *Comparaison entre les formats d'image*

I.10. Fenêtrage

En traitement du signal, le fenêtrage est utilisé dès que l'on s'intéresse à un signal de longueur volontairement limitée. En effet, un signal réel ne peut qu'avoir une durée limitée dans le temps ; de plus, un calcul ne peut se faire que sur un nombre fini de points. **[27]**

I.10.1 Fenêtres courantes

- ***Fenêtre de Hamming***

$$w[n] = 0.54 + 0.46\cos\left(2\pi * \frac{n}{2M+1}\right) \qquad \textbf{(I.8)}$$

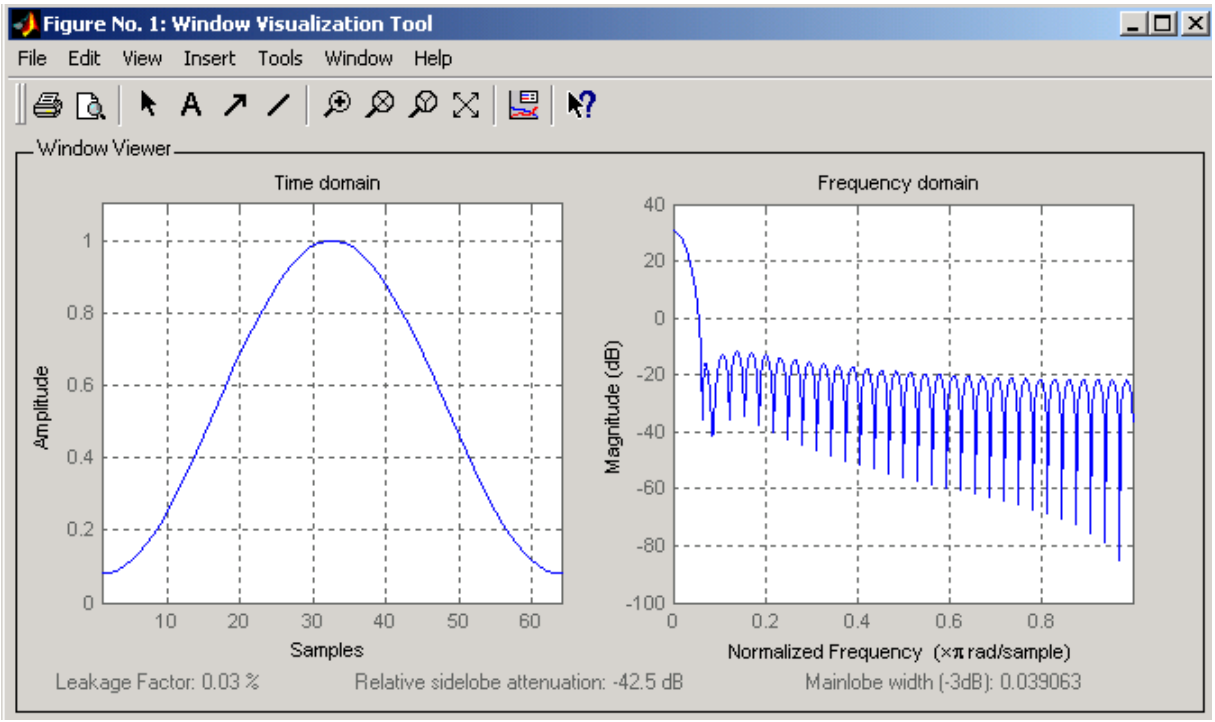


Figure I.16. *Fenêtre de Hamming (256 samples à 250 kHz, fc=10 kHz)*

- Bande de transition étroite (inférieure à 1 kHz).
- Atténuation stagnante dans la bande rejetée (-55 dB)

- ***Fenêtre de Hanning***

$$w[n] = 0.5 + 0.5\cos\left(2\pi * \frac{n}{2M+1}\right) \qquad \textbf{(I.9)}$$

La figure suivante représente l'évolution de la fenêtre de hanning :

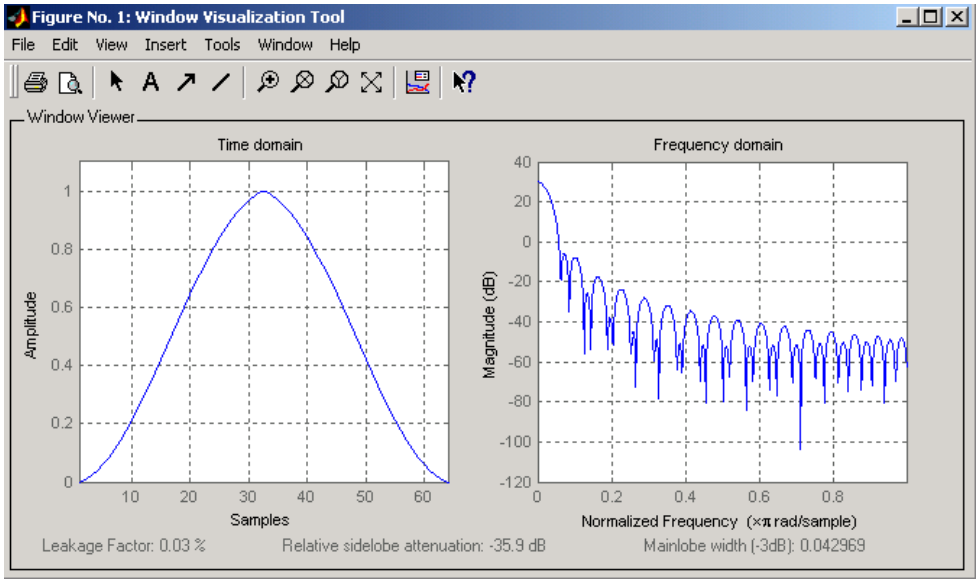


Figure I.17.*Fenêtre de Hanning (256 samples à 250 kHz, Fc=10 kHz)*

- Atténuation stagnante dans la bande rejetée (-75 dB)
- Bande de transition étroite (inférieure à 2 kHz).

- ***Fenêtre de Blackman***

$$w[n] = 0.42 + 0.46\cos\left(2\pi * \frac{n}{2M+1}\right) + 0.08\cos\left(2\pi * \frac{n}{2M+1}\right) \quad \text{(I.10)}$$

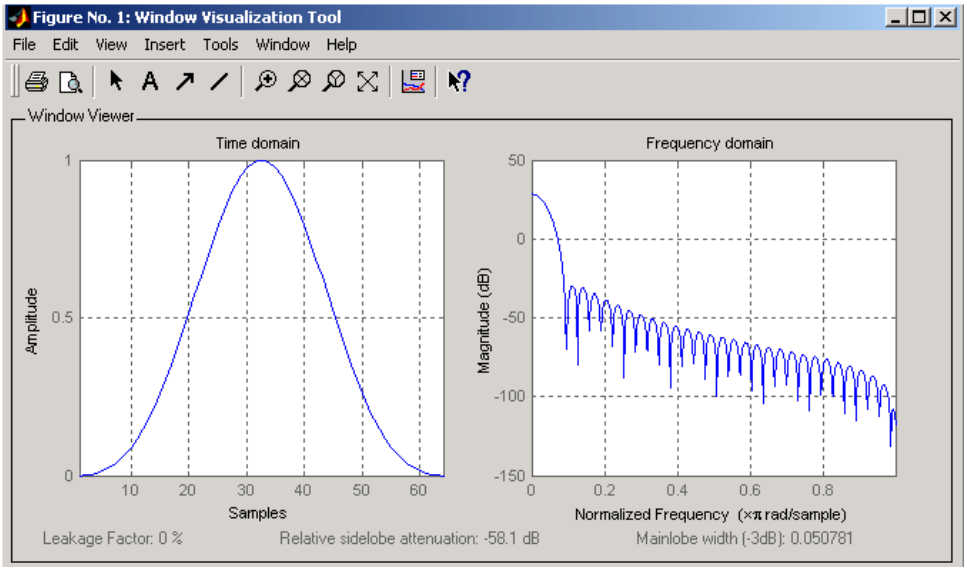


Figure I.18. *Fenêtre de Blackman (256 samples à 250 kHz, Fc=10 kHz)*

- Cette atténuation en bande rejetée.
- Faible bande de transition (2,5 kHz pour atteindre 60 dB d'atténuation contre 2 kHz pour la fenêtre de Hanning).

Remarque

Chacune d'entre elles possède leurs propres avantages et inconvénients. Par exemple, si l'on veut un filtre ayant une faible bande de transition, peu importe l'atténuation, il est préférable d'utiliser une fenêtre Rectangulaire. En revanche, s'il est question d'une atténuation minimale de 30 dB, il vaut employer une fenêtre de Hanning. La fenêtre de Blackman correspond aussi à ce gabarit du filtre mais sa bande de transition est trop importante.

Il y a aussi la fenêtre qui est très courante dans ces dernières années s'appelle ondelette.

- **Fenêtre Ondelette**

a- Qu'est-ce qu'une ondelette?

L'ondelette c'est une petite onde (ou vague) qui a un début et une fin. Est utilisé pour représenter une fonction (ou un signal) comme une somme pondérée de ces petites ondes translatées ou dilatées. **[23] [24]**

Les fonctions analysantes ou ondelettes sont comme les sinus et les cosinus, qui permettent de décomposer un signal. Ils sont localisés en temps et en fréquence. Le caractère localisé de l'ondelette s'exprime par le fait que la fonction est non nulle sur un intervalle fini et nulle partout ailleurs.

$$\psi_{a,b}(t) = a^{-1/2}\psi\left(\frac{t-b}{a}\right) \qquad \text{(I.11)}$$

Avec :

$\boldsymbol{\psi(t)}$ *:* L'ondelette mère (fonction de fenêtrage qui possède deux coefficients).

a : l'échelle (coefficient de dilatation de l'ondelette égale à1/f).

b : coefficient de translation (déplacement de la fenêtre le long du signal).

Les ondelettes sont des dilatations et des translations d'une ondelette initiale que l'on nomme ondelette mère.

***b- quelques types d'ondelettes utilisés :* [25] [26]**

✓ **Le chapeau Mexicain**

L'ondelette appelée chapeau mexicain s'obtient par double dérivation de la gaussienne $e^{-\frac{t^2}{2}}$

Son expression normalisée en énergie s'écrit :

Sa définition: mexh (x) = c*exp (-x2/2) * (1-x2)

Ou : c = 2/(sqrt (3)* pi ¼)

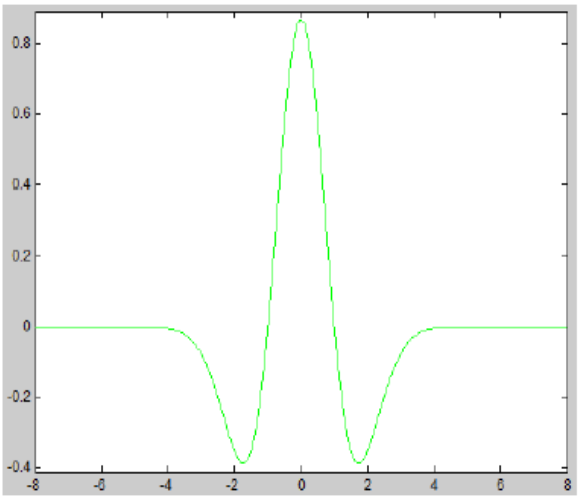


Figure I.19. *Ondelettes Chapeau Mexicain*

✓ **L'ondelette de Meyer**

Cette ondelette est basée sur une fonction v(t) réelle paire et indéfiniment dérivable, l'ondelette de Meyer est définie par son intégrale :

$$\psi(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} cos\left(t - \frac{1}{2}\right) * \sin[\upsilon(t)] \qquad \text{(I.12)}$$

$$\begin{cases} \upsilon(t) = 0 \quad \text{pour } 0 \le t \le \dfrac{2\pi}{3} \\ \upsilon(t) = 0 \quad \text{pour } t \ge \dfrac{8\pi}{3} \\ \upsilon(2t) = \dfrac{\pi}{2} - \upsilon(t) \quad \text{pour } t \in S \\ \upsilon(t - 2p) = \dfrac{\pi}{2} - \upsilon(t) \quad \text{pour } t \in S \end{cases}$$

D'où **S** est l'intervalle $\left[\frac{8\pi}{3}, -\frac{8\pi}{3}\right] \cup \left[\frac{2\pi}{3}, -\frac{8\pi}{3}\right]$

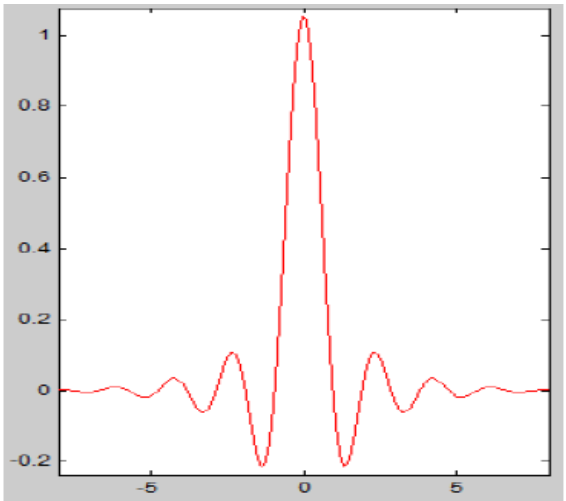


Figure I.20. *Ondelettes de Meyer*

- ✓ **Ondelettes de Morlet**

C'est une gaussienne modulé :

$$\psi(t) = e^{-\pi t^2} e^{2i\pi K_0 t}$$

Sa définition: morl (x) = exp(-x2/2) * cos (5x).

Sa transformée de Fourier est gaussienne décalé de k0 et vaut :

$$\hat{\psi}(w) = e^{-\pi(w-K_0)^2}$$

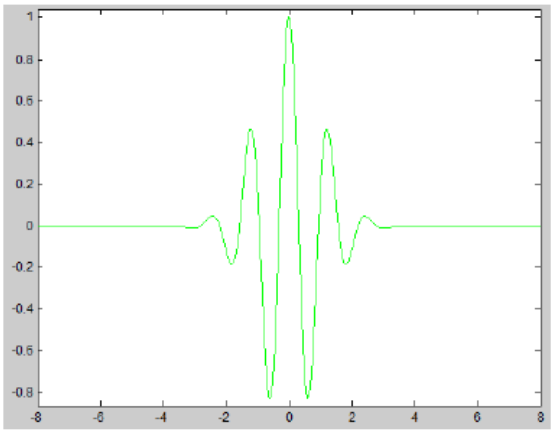


Figure I.21. *Ondelettes de Morlet*

I.11. Conclusion

Dans ce chapitre, nous avons décrit brièvement quelques notions de base sur l'image numérique et son traitement comme: l'échantillonnage, la quantification et filtrage, finalement le seuillage pour obtenir un bon résultat.

Ainsi nous avons donné un aperçu générale les types des fenêtres courante (Hamming, Hanning, Blackman, Ondelettes) qui nous devons les utiliser dans notre projet.

Dans le domaine de traitement d'image numérique on cherche à détecter la présence de certaines formes, certains contours ou certaines textures d'un modèle connu, c'est le problème de la détection.

Désormais ; nous pouvons aborder le chapitre suivant qui concerne les méthodes de détection de contour d'images en présentant, quelques méthodes classiques de détection de contour citons par exemple la méthode de CANNY et la méthode de DERICHE.

CHAPITRE II

Techniques de détection de contours d'image

Chapitre II

Techniques de détection de Contours d'image

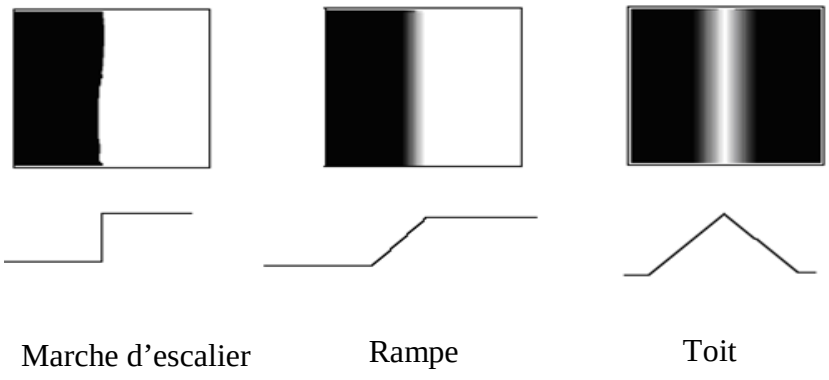
II.1 Introduction

La détection de contours n'est pas un objectif en soi, elle est la première étape dans un système de vision. En effet, un détecteur de contours fournit en résultat un ensemble de points. Ceux-ci sont regroupés pour former des chaînes de points de contours qui peuvent être considérées comme une représentation intermédiaire. La quantité des résultats et le temps de traitement de ce système dépendent directement ou indirectement de la qualité de cette représentation intermédiaire.

Le problème qui se pose, c'est que le but recherché (contour) ne peut pas être défini avec précision. Ainsi, il n'existe pas d'algorithme unique donnant satisfaction dans le cas général. Cette constatation fait dire à plusieurs auteurs quel problème est mal conditionné, en pratique il existe autant d'algorithmes que de contours**.**

II.2 Le contour

Une arête (contour) est une ligne visible dans l'image, située à la séparation entre deux zones de l'image. L'arête correspond à une frontière d'un objet présent dans l'image. Le long de l'arête, le niveau de gris change peu, par contre il change fortement dans une coupe perpendiculaire à celle-ci. Par exemple, une arête verticale correspond à une variation rapide des niveaux de gris dans la direction horizontale.



FigureII.1 *quelques modèles de contours. Le plus utilisé est celui en marche d'escalier.*

II.3 Principe général de détection de contours

D'un point de vue conception, un détecteur inclut trois opérations: le lissage, la différentiation et la correction.

a. Le lissage

Le lissage consiste à réduire le bruit contenu dans l'image. Cependant, le lissage a pour effet indésirable d'éliminer et de déplacer certains contours et de créer de nouveaux contours. Il est clair qu'un filtre de lissage doit assurer un compromis entre l'élimination de bruit et la conservation des contours. Il importe de souligner que l'étape de lissage a différentiation numérique de l'image.

b. La différentiation

La différentiation consiste à calculer les dérivées nécessaires qui permettent de mettre en évidence les contours. Un opérateur de différentiation est caractérisé par son ordre, son invariance par rotation et par sa linéarité. Les opérateurs différentiels les plus utilisés en détection de contours sont:

- ✓ Les opérateurs différentiels directionnels(DD) dont le gradient.
- ✓ Les opérateurs différentiels invariants par rotation (RID) dont on cite: le laplacien
- ✓ La dérivée seconde prise le long de la direction du gradient**: [1], [3].**

c. La correction

La correction a pour objectif d'augmenter le rapport: signal bruit du détecteur ainsi que la localisation des contours par une suppression des faux points de contours par la fermeture des trous dans les contours **[2], [6].**

II.4Techniques linéaires de détection de contours

La détection de contours s'effectue de deux manières : l'approche du gradient et l'approche du Laplacien comme le montre la figure II.2 suivante:

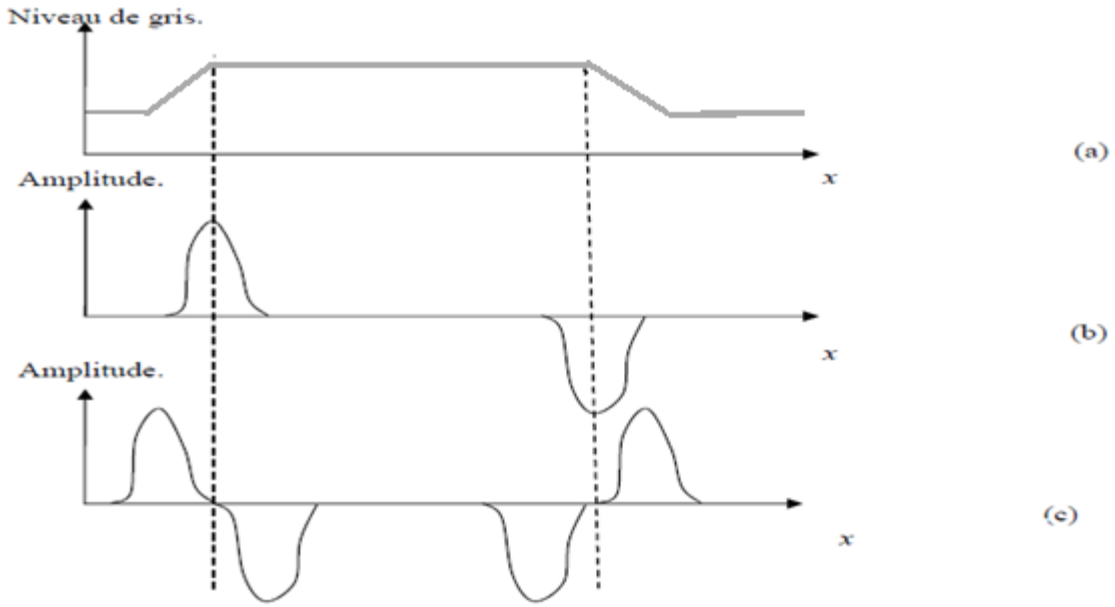


Figure II.2 *détection de contour (dérivation) : (a) : Contour (rampe), (b) : Gradient, (c) : Laplacien.*

II.4.1Approche gradient

C'est le calcul du gradient et extraction des extrema locaux de la norme du gradient dans la direction du gradient, ce qui ramène à extraire les zéros de la dérivée seconde dans la direction du gradient.

Calcul du gradient: (voir figure II.2.b).

$$\vec{\nabla} I = \frac{\Delta l}{\Delta x}\vec{\imath} + \frac{\Delta l}{\Delta y}\vec{\jmath} = G\,(x, y) \qquad (II.1)$$

Le vecteur $\vec{\nabla}$ est orienté dans le sens de variation maximale de *I*, l'extraction des points de contours s'effectue par sélection des points de norme de gradient élevée grâce aux deux étapes suivantes:

a. Calcul du la norme du gradient

$$N\,(x,y) = \left| (G_x\,(x,y))^2 + (G_y\,(x,y))^2 \right|^{1/2} \qquad (II.2)$$

b. Sélection des points de fort gradient

N(x,y) > s ,seuil fixe.

Il n'existe pas de seuil (*s*) permettant d'obtenir les vrais points de contours sans sélectionner aussi ceux dû au bruit.

Un moyen est d'extraire les extrema locaux de la norme du gradient dans la direction du gradient. Une méthode efficace consiste à:

- Extraire les extrema locaux de gradient.
- Réaliser un seuillage par hystérésis des extrema.

b.1. Extraction des extrema locaux du gradient

Ces extrema correspondent au passage par zéro de la dérivée seconde dans la direction de gradient. Soient: M*(x,y)*:un point de gradient *G(M).*
M1(x,y),M2(x,y):deux points de la droite passant par *M* et de vecteur directeur *G(M).*

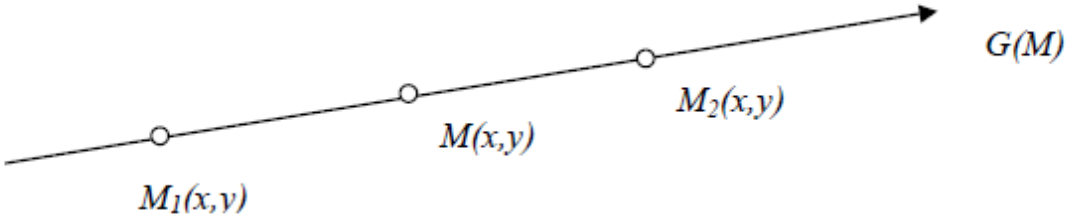


Figure II.3 *Extraction des extrema locaux.*

Le point *M* est sélectionné si :*N(M)>N(M2) et N(M)≥N(M1).*

(*N*: représente la norme du point considéré)

L'extraction des extrema locaux revient à se déplacer le long de la normale au contour (approximation par gradient) et à détecter le point de plus fort gradient.

b.2 Seuillage par hystérésis des extrema

Ce seuillage permet d'éliminer les points de norme de gradient faible, et nous donne des points de contours bien connectés entre eux.

II.4.2. Approche Laplacien

C'est le calcul de la dérivée seconde d'une fonction (image), qui permet de donner la

valeur de Laplacien en chaque point de l'image, d'où les points de contours sont localisés au passage par zéro du Laplacien.

a- Calcul de Laplacien : (VoirfigureII.2(c)).

Soit une image *I (x,y)*:

$$\nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \qquad \text{(II.3)}$$

D'après la **figureII.2** (c) on peut localiser les passages par zéro du laplacien et signaler à chaque passage, la présence d'un contour par un point.

b- Étapes d'extraction de passage par zéro

b.1 Détermination d'une image de polarité, en calculant une image I_p telle que

$$I_p(M) = \begin{cases} 1 & si \quad lap(M) \leq 0 \\ 0 & si \quad lap(M) > 0 \end{cases}$$

M : un point de coutour

b.2 Détermination (détection) des passages par zéro, en calculant une image *Iz* telle que

$$I_p(M) = \begin{cases} 1 & si \;\; M\ correspond \text{ à } une\ transition\ (0 \text{ à } 1\ ou\ 1 \text{à } 0) \\ 0 & si\ non \end{cases}$$

b.3 Seuillage des passages par zéro

L'élimination des passages par zéro de faible norme de gradient peut s'effectuer par un algorithme de seuillage quelconque. L'algorithmede seuillage par hystérésis décrit pour l'approche dérivée première peut par exemple être utilisé.

II.5 Différentes étapes de la détection de contours

Soit $I(\vec{\imath}, \vec{\jmath})$ une image d'entrée.

a. Choix des filtres de lissage

Nous nous intéressons particulièrement aux filtres séparables et récursifs car ils peuvent se générai à une dimension quelconque et permettent une implantation de faible coût algorithmique **[4].**

Parmi les filtres de lissage, on peut citer:

- Filtre de DERICHE.
- Filtre de CANNY.
- Filtre Paramétrique .

b. Choix de type d'approche

En général on constate que le calcul de la dérivée seconde est plus sensible au bruit. D'un autre côté, la complexité du calcul du gradient est élevée pour le laplacien. Ceci est dû aux simplifications qui apparaissent lorsqu'on calcule la réponse du filtre d'approximation du laplacien par multiplication et addition de filtre de lissage et de dérivée seconde. Il faut cependant noter que utile et qu'elle nécessite le calcul de la norme du gradient (aux points de passage par zéro). La localisation des contours obtenus est pratiquement (exactement) la même. **[9]**

c. Détermination d'une cascade de filtres pour approximer le gradient ou le laplacien

Une fois que l'on a choisi le type d'approche: gradient ou laplacien, il faut déterminer un moyen de les calculer en introduisant des opérateurs permettant d'approximer ce calcul.

c.1 Opérateurs de Sobel et Prewitt

Les opérateurs de Sobel et de Prewitt permettent d'estimer localement la norme du gradient spatial bidimensionnel d'une image en niveau de gris. Ils s'amplifient les régions de fortes variations locales d'intensité correspondant aux contours. Ces opérateurs consistent en une paire de masques de convolution 3×3. L'application séparée de chacun des masques donne une estimation des composantes horizontales et verticales du gradient par un simple filtrage linéaire avec un masque 3×3. Il est ensuite possible de calculer la norme et la direction du gradient en chaque point à partir des composantes du gradient ∇xet ∇y. La norme du gradient ainsi estimée correspond à l'intensité attribuée au pixel courant. C'est donc l'image de la norme du gradient que l'on visualise généralement.

Les opérateurs de Sobel et de Prewitt combinent un léger filtrage passe-bas

le long des lignes et des colonnes et un filtrage passe-haut du type gradient (changement de signe des poids). Nous donnons ici les masques de Sobel et de Prewitt:**[10]**

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \qquad P_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix},$$

$$S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \qquad P_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix},$$

Masque de sobel. Masque de Prewitt

c.2 Opérateur de Robert

Le détecteur de Roberts permet de calculer le gradient bidimensionnel d'une image de manière simple et rapide. Il amplifie les zones où la norme du gradient spatial est importante qui correspond souvent aux contours. Ce principe ne diffère pas beaucoup de celui des opérateurs de Prewitt et Sobel. Ces masques répondent maximalement aux contours orientés à 45° par rapport à la grille de pixels. Par conséquent, l'opérateur de Robert s'effectue une estimation locale du gradient selon les deux diagonales.**[10]**

$$R_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, R_x = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Masque de Robert

II.6 Les méthodes classiques de détection de contours

Une des difficultés de détection de contours provient du bruit important présent dans les images (bruit de capteur, bruit d'échantillonnage, irrégularités de la surface des objets,…). On est donc confronté au problème de la différentiation d'un signal bruité. Pour le résoudre il est d'abord nécessaire de:

- ✓ Définir des critères de performance d'un opérateur incluant une modélisation des contours recherchés et la notion de bruit.
- ✓ Déduire de ces critères une famille de filtres optimaux.
- ✓ Résoudre le problème posé par l'implantation de ces filtres.

II.6.1 La méthode de CANNY

Dans toutes les méthodes de détection de contours, il existe des critères communs concernant la performance du détecteur. **[3].**

CANNY proposa de quantifier les performances d'un détecteur de contours par les trois critères suivants**:**

- La détection : l'opérateur doit donner une réponse au voisinage d'un contour.
- La localisation : le contour doit être localisé avec précision.
- Réponse unique: un contour doit provoquer une seule réponse de l'opérateur de détection.

Le problème de la détection est formulé par CANNY comme suit :

- ✓ Nous commençons avec un contour ayant une «cross section» connue, noyée dans un bruit blanc gaussien.
- ✓ CANNY propose de marquer le centre d'un contour par un maximum local dans la sortie de la convolution de l'image avec le filtre que nous recherchons et qui doit satisfaire aux trois critères suivants:

a. Bonne détection

La qualité de détection est déterminée par une faible probabilité de ne pas omettre un vrai point de contour et de ne pas marquer contour un point image qui ne le soit pas. Ce critère correspond à maximiser le rapport signal bruit SNR de la réponse du détecteur.

Soit *I(x)* l'image définie par **:** *I(x)=f(x)+N(x)* où *f(x)* est le signal non bruité (le contour) et *N(x)* le bruit blanc gaussien. Soit *f(x)=AU(x)* le profil d'une marche idéal où *A* est l'amplitude de la marche.

$$U(x)\begin{cases}0 & Si \quad x<0\\ 1 & Si \quad x>0\end{cases}$$

SNR= $(A/\eta 0)$. $\sum(f)$ $\eta 0$: variance de bruit blanc

$$\text{Où : } \sum(f)=\frac{\left|\int_{-w}^{0} f(x).dx\right|}{\sqrt{\int_{-w}^{+w} f'^{2}(x).dx}} \qquad \text{(II.4)}$$

ω: étant la taille du filtre défini dans *[0,ω]*.

b. Bonne localisation

Pour la localisation, CANNY cherche une mesure qui augmente au fur et mesure que la localisation s'améliore. Pour cela, il propose d'utiliser l'inverse de la distance entre le maximum de la réponse et la position réelle du contour.

Puisqu'il décide de marquer les contours aux maximaux locaux dans la réponse de l'opérateur f (x), la première dérivée de la réponse sera nulle en ces points.

Notons aussi que puisque les contours sont centrés en *x=0*, en absence du bruit, le maximum local sera en *x=0*. Alors nous avons:

Localisation $\approx \frac{A}{\eta O}.A(f') = A/(\lambda.\eta_0)$;

$$\lambda = \frac{\sqrt{\int_{-w}^{+w} f'^2(x).dx}}{|f'(0)|} \qquad \text{(II.5).}$$

$$\text{Où } A\left(f^{'}\right) = \frac{\left|f^{'}(0)\right|}{\sqrt{\int_{-w}^{+w} f^{'2}(x)dx}} \qquad \text{(II.6).}$$

c. Réponse unique

Le détecteur ne doit pas fournir de multiples réponses à un seul contour, la distance entre les pics dans la réponse au bruit de *f, notée* : *X max* est posée comme une fraction *K* de la longueur de l'opération ω.

$$X_{max}(f) = K.w = 2.\pi.\frac{\int_{-\infty}^{+\infty} f'^2(x).dx}{\int_{-\infty}^{+\infty} f''^2(x).dx} \qquad \text{(II.7)}$$

Ces critères sont combinés d'une manière significative: on maximise le produit *(∑.λ)* sans la contrainte du 3ième critère. On obtient ainsi une équation différentielle dont la solution est:

$$F(x)=a_1 e^{\alpha.x}.\sin\omega x + a_2.e^{\alpha x}\cos\omega x + a_3 e^{-\alpha.x}\sin\omega x + a_4 e^{-\alpha x}\cos\omega x + c \qquad (II.8)$$

avec: $f(0)=0; f(w)=0 ; f'(0)=s ; f(-w)=0.$

s: la pente de *f(x)* à l'origine.

Par une optimisation numérique, CANNY détermina qu'un compromis optimal pour un filtre *RIF*. Pour une mise en pratique concrète, l'implémentation se réalise à l'aide de la dérivée d'une gaussienne, on obtient alors $\sum A=0.92$ et $X_{max}=0.51$.

II.6.1 La méthode de DERICHE

CANNY propose un filtre à réponse impulsionnelle finie (RIF), alors que DERICHE propose un filtre à réponse impulsionnelle infinie (RII) satisfaisant les mêmes critères. La solution générale étant toujours la même **[3]** :

$$F(x)=a_1 e^{\alpha.x}.\sin\omega x + a_2.e^{\alpha x}\cos\omega x + a_3 e^{-\alpha.x}\sin\omega x + a_4 e^{-\alpha x}\cos\omega x + c.$$

Seules les conditions aux limites changement:

$f(0)=0 ; f(-\infty)=0 ; f'(0)=s ; f(+\infty)=0.$

En appliquant ces conditions à l'équation précédente nous obtenons:

$$\boldsymbol{f(x)=\frac{s}{\omega}e^{-a|x|}\sin(\omega x)} \qquad (II.9).$$

$$\boldsymbol{A=\sqrt{2\alpha}} ; \sum = \sqrt{\frac{2\alpha}{a^2+\omega^2}} \quad \text{et} \quad \boldsymbol{X_{max}} = \sqrt{\frac{\alpha^2}{5\alpha^2+\omega^2}} \qquad (II.10)$$

Ce filtre répond exactement aux mêmes critères de qualité que celui de CANNY.

DERICHE a déterminé qu'en prenant α très grand devant F on obtient des résultats optimaux : $\sum A = 2$ et $Xmax = 0.44$.

II.7Conclusion

Dans ce chapitre, nous avons présenté les différentes méthodes de détection de contour d'image fixe, qui sont les méthodes les plus utilisées dans le traitement d'image.

Le choix d'une méthode est lié à plusieurs facteurs comme: le type de l'image (N &B, RGB,…), temps d'exécution, sensibilité au bruit et bien évidemment les contraintes d'exploitations.

Du fait de cette diversité, il est difficile de définir, de manière absolue, une bonne méthode de détection de contour. Parmi les méthodes les plus modernes, on trouve celles du filtrage paramétrique, qui devient plus sensible au bruit, et c'est ce qui sera étudié dans le chapitre suivant.

CHAPITRE III

Détection de contour par la méthode PF

Chapitre III

Détection de contours par la méthode du filtrage paramétrique

III.1 Introduction

Les nouvelles techniques de traitements du signal **[16] [17]** pour détecter les contours d'images sont considérables. L'une de ses approches est appelée filtrage paramétrique. Cette méthode est appliquée avec succès sur la parole et la segmentation vocale, on va la projeter par application sur la détection du contour.

Cette nouvelle approche ouvre des voies très prometteuses dans l'analyse de contour et la segmentation, la méthode proposée combine un filtre paramétrique avec l'analyse de l'autocorrélation du premier ordre du signal image filtré, ce fait produira une nouvelle fonction caractéristique pour le spectre de l'image, en se basant sur cette nouvelle fonction caractéristique, divers mesures de distorsion sont proposées comme des indicateurs du changement spectral.

Ces indicateurs présentent de bonnes performances pour la détection de contour et pour la segmentation.

III.2 Vue générale sur la méthode du filtrage paramétrique

Il y a plusieurs méthodes dans la littérature de détection de contour d'image basées sur différents techniques de détection, les uns appliquent la dérivée première (le gradient), d'autres appliquent la dérivée seconde (laplacien).

Dans cette section nous allons brièvement revoir quelques procédures qui sont sans contraintes (dépendant seulement des informations sur l'objet), et celles qui conviennent le mieux à notre méthode, notamment la segmentation par la détection des variations spectrales du signal. Pour simplifier, nous classons les méthodes dans deux catégories : les méthodes à modèles et les méthodes sans modèles **[18] [19].**

III.2.1 Les méthodes à modèle

Elles dépendent essentiellement des modélisations LPC (**Linear Prodiction Coding**) qui due à des considérations de l'efficacité des calculs.

Une des méthodes est dérivée d'un test GLR (**Generalized Liklihood Ratio**) en supposant que le signal entrant est une partie d'un processus autorégressif conduite par le bruit blanc Gaussien. En donnant un domaine du signal, la méthode, en principe, teste l'hypothèse qu'aucun changement ne se produit dans le domaine alors qu'une variation a lieu dans ce domaine à un certain temps t.

En utilisant les modèles LPC, les fonctions de probabilités Gaussienne (Gaussien liklihood functions) sont évaluées sous les deux hypothèses et le rapport de vraisemblance (the liklihood ratio) D_t est calculé pour chaque t. Si D_t est maximisé en t=t* et le maximum excède un certain seuil, un changement en t=t* est apparu.

En plus, les rapports de vraisemblance des modèles LPC sont aussi employés comme estimateurs spectraux (Spectral Estimators) dans le calcul des mesures de distorsion pour détecter les variations spectrales.

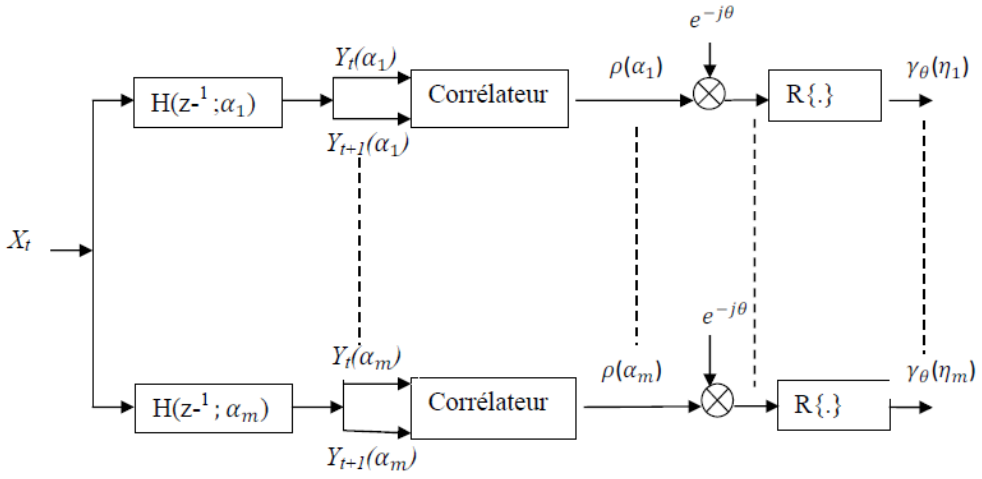


Figure III.1*schéma bloc de filtrage paramétrique.*

III.2.2 Méthodes sans modèles

Les méthodes sans modèles existantes sont généralement basées sur l'utilisation directe du spectre de l'image, couplée avec une mesure de distorsion spectrale. Par exemple Deshayes et Picard utilisent la distance spectrale de **Kolmogorov-Smirnov** :

$D = sup|F_1(\omega) - F_2(\omega)|$est la distance spectrale de **Cramer-Von**:

$D = \int |F_1(\omega) - F_2(\omega)|^2 d\omega$ où $F_1(\omega)\ et\ F_2(\omega)$sont des fonctions de distributions spectrales estimées des deux domaines du signal pris dans le voisinage du point t. La variation est déclarée comme avoir lieu en t si la distance calculée en t excède un seuil.

D'autres distances spectrales (mesures de distorsions) peuvent être aussi bien employées pour ces buts, deux exemples sont :

- La divergence spectrale de Kullback-Leibler (KL).

$$D_{KL} = \int_{-\pi}^{\pi} K\left[\frac{f_1(\omega)}{f_{2(\omega)}}\right] d\omega \qquad \text{(III.1)}$$

Où *K(u) = u –log (u) – 1*, tel que *K* est la distance de Kulback-Liebler.

- Et la distance L_2 de log-spectrum :

$$D_{LL2} = \int |\log f_1(\omega) - \log f_2(\omega)|^2\ d\omega \qquad \text{(III.2)}$$

Où : $f_1(\omega) et\ f_2(\omega)$sont des fonctions de densité spectrale estimées de deux domaines du signal au voisinage de' t'.

III.3 La Méthode de Filtrage Paramétrique (PF)

La méthode de filtrage paramétrique (PF) est motivée par le fait que la structure de corrélation d'un signal stationnaire peut être caractérisée par la signature de certaines sorties statistiques d'un filtre judicieusement désigné**. [17][22]**

III.3.1 Méthode de base

Supposons $\{X_t\}$, est un signal stationnaire à valeurs réels avec la moyenne égale à zéro et une fonction d'autocorrélation égale à :

$$\rho_k = \frac{E(X_{T+K} * X_T)}{E(X_t^2)} \qquad \text{(III.3)}$$

Considérons le filtre récursive IIR H (Z^{-1}; a) définie par :

$$Y_t(\alpha) = \sum_{l=0}^{\infty} \overline{\alpha^l} X_{t-1} = \bar{\alpha} Y_{t-1}(\alpha) + X_t \text{(III.4)}$$

Où : $\alpha = \eta e^{-i\theta}$ .est un nombre complexe avec $|\eta| < 1\ et\ \ \theta \epsilon [-\pi, \pi]\ et\ \bar{a}$est le conjugué.

Soit ρ(α) l'autocorrélation d'ordre un de $\{Y_t(\alpha)\}$

$$\rho(\alpha) = \frac{E\left(Y_{t+1}(\alpha).\overline{Y_t}(\alpha)\right)}{E(|Y_t(\alpha)|)^2} \quad \text{(III.5)}$$

Pour tout θ fixé, on définit l'autocorrélation de premier ordre démodulée du signal $\{Y_t(\alpha)\}$ comme suit :

$$\gamma_\theta(\eta) = \Re\left(e^{-i\theta}\rho(\alpha)\right) \text{(III.6)}$$

Où : $\Re(.)$ représente la partie réelle d'un nombre Complexe.

Dans ce projet, on utilise $\gamma_\theta(\eta)$ comme une nouvelle fonction caractéristique, complémentaire au spectre de Fourier pour la représentation de la structure de corrélation de $\{X_t\}$.Nous appellerons cette méthode d'analyse des structures d'autocorrélation « **laméthode du filtrage paramétrique** ».

Pour calculer$\gamma_\theta(\eta)$, nous n'allons imposer aucun modèle paramétrique ou des suppositions de distributions sur$\{X_t\}$, c'est pour ça que cette méthode appartient à la catégorie des méthodes sans modèles sans utiliser explicitement les densités spectrales.

Dans l'application, cette méthode peut être implémentée facilement avec un filtre paramétrique de paramètre α comme le montre la figure (III.1), ou les $\eta_k$peuvent être pris par exemple, uniformément dans l'intervalle$[\eta_\alpha, \eta_b] \subset [-1,1]$.

Pour chaque évaluation de$\gamma_\theta(\eta)$, le nombre de multiplications exigé est proportionnel à la longueur du signal appliqué, et habituellement, quelques évaluations suffisent pour les besoins de la détection de contour proposée.

III.3.2 Mesures de distorsion

Étant donné deux domaines (ou plages) du signal image à traiter, nommés$X_t^{(1)}$et$X_t^{(2)}$, comme deux fenêtre à traiter, la propriété caractéristique $de\ \gamma_\theta(\eta)$ peut être utilisée pour dériver des mesures de distorsion qui quantifient les déviations de $X_t^{(1)}$ et$X_t^{(2)}$ dans leurs structures de corrélation. **[18]**

Pour la segmentation d'images,$X_t^{(1)}$ et$X_t^{(2)}$ peuvent être pris comme des segments donnés sous forme de pixels de différentes intensités (niveau de gris), ainsi de grandes valeurs de distorsion indiquent la présence significative de changements dans la structure de corrélation.

La méthode du filtrage paramétrique appliquée à la détection de contour d'une image est basée sur des mesures de distorsion qui incluent la distance $L_p$de $\gamma_\theta(\eta)$, c'est-à-dire :

$$\gamma_\Omega^p = \left\{ \int_\Omega \left| \gamma_\theta^{(1)}(\eta) - \gamma_\theta^{(é)}(\eta) \right|^P . d\theta . d\eta \right\}^{1/P} \qquad \text{(III.7)}$$

Pour $p \in]0, \infty[$, et la distance KL définie par :

$$\widehat{K_\Omega} = \int_\Omega \left(K\left[\frac{P_\theta^{(1)}(\eta)}{P_\theta^{(2)}(\eta)}\right] + K\left[\frac{P_\theta^{(2)}(\eta)}{P_\theta^{(1)}(\eta)}\right] \right) d\theta d\eta \qquad \text{(III.8)}$$

Dans cette expression Ω est un sous ensemble de $[-\pi, \pi] * [\eta_\alpha, \eta_b]$, $\gamma_\theta^{(1)}(.) et\, \gamma_\theta^{(2)}(.)$: sont des fonctions caractéristiques obtenues de $X_t^{(1)}$ et$X_t^{(2)}$ respectivement, Et $p_\theta^{(1)}(.) et\, p_\theta^{(2)}(.)$sont des fonctions de densités de probabilité définies dans l'intervalle$[\eta_\alpha, \eta_b] \subset [-1,1]$, ayant la forme :

$$P_\theta(\eta) = \frac{1}{2}\left(\frac{d\gamma_\theta(\eta)}{d\eta} + [\gamma_\theta(\eta_a) + 1]\delta(\eta - \eta_a) + [1 - \gamma_\theta(\eta_b)]\delta(\eta - \eta_b \right) \qquad \text{(III.9)}$$

Notons que$P_\theta(\eta)$ possède aussi la propriété caractéristique à cause de son équivalence à $\gamma_\theta(\eta)$. Sachant que :

$$\int_{\eta_a}^{\eta_b} P_\theta(\eta).d\eta = 1\ et\ \gamma_\theta(\eta) = 2.\int_{\eta_a}^{\eta_b} p_\theta(\eta).d\eta - 1\ pour\ tout\ \eta\ \in\ [\eta_a, \eta_b]$$

Dans la pratique, ces mesures de distorsion sont discrétiser en utilisant la sortie $\gamma_\theta(\eta_k)$, du bloc de filtre dans la figure (III.1).

III.4 Présentation d'algorithme

L'algorithme se compose d'un ensemble de blocs :

III.4.1 Bloc de filtrage (le lissage)

Ce filtre et caractérisé aussi par une réponse impulsionnelle infinie « IIR » et doit répondre à quelque contraintes :

1. La séparabilité, pour réduire la complexité du calcul.
2. Pour ne pas introduire un bruit de repliement de spectre, la réponse en fréquence associée doit être nulle en dehors du domaine de NYQUIST. Cette

contrainte ne pourra pas être satisfaite car nous implémentons un filtre paramétrique de paramètre complexe $\alpha \in (.)$.

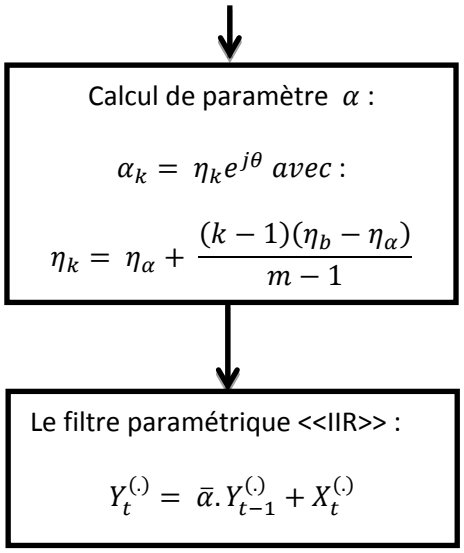
3. Le contour doit être localisé avec précision.

Nous implémentons notre filtre paramétrique de manière récursive pour deux raisons :

a) L'efficacité.

b) L'optimisation de l'erreur d'approximation.

Ce bloc du filtrage se déroule comme suit :



On prend deux signaux extraits de chaque ligne de l'image et qui sont sous forme des fenêtres de hamming (hamming window) : $X_t^{(1)}$ et$X_t^{(2)}$ont une largeur de N pixels et sont intercalées de M = N/2 (exemple : N=8).

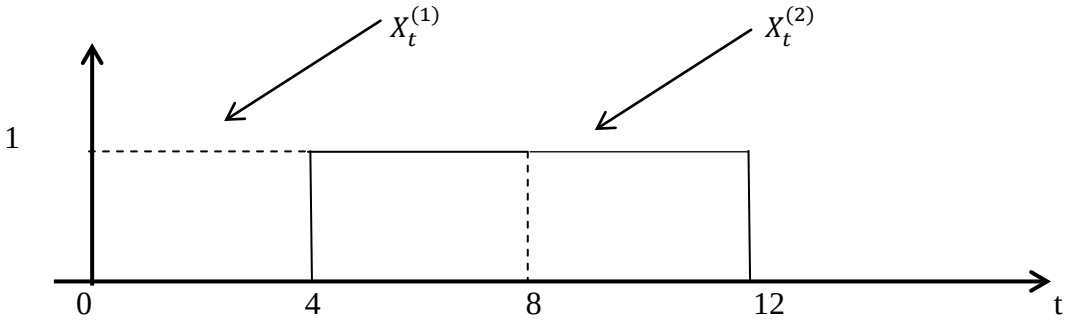


Figure III.2*représentation des deux portes décalées de 50 % de chevauchement.*

Ensuite, on filtre les deux signaux : $X_t^{(1)}$ et$X_t^{(2)}$ par notre filtre paramétrique $H\ (z^{-1}, \alpha)$ sous forme récursive et est calculé avec θ fixe et m spécifie l'espace de η_k dans l'intervalle $[\eta_\alpha, \eta_b]$ tel que :

$$\eta_k = \eta_\alpha + \frac{(k-1)(\eta_b - \eta_\alpha)}{m+1} \qquad pour\ k = 1, \ldots\ldots, m \quad \text{(III.9)}$$

Ainsi, en suivant une pratique commune dans la détection de contour et la reconnaissance des formes, on applique ce filtre paramétrique comme suit :

$$Y_t^{(.)} = \bar{\alpha}.Y_{t-1}^{(.)} + X_t^{(.)}\ ,\ ou:\ \overline{\alpha_k} = \eta_k e^{-j\theta} \quad \text{(III.10)}$$

D'après la transformée en Z, on trouve que : $H(z) = \frac{Y_t}{X_t} = \frac{1}{1-\bar{\alpha}.z^{-1}}$. (III.11)

III.4.2 Bloc d'autocorrélation du premier ordre

On prend les deux signaux filtrés:$Y_t^{(1)}$ et$Y_t^{(2)}$et on calcule leur autocorrélation de premierordre par l'expression suivante :

$$\rho(\alpha) = E\left\{Y_{t+1}^{(.)}(\alpha).\bar{Y}_t^{(.)}(\alpha)\right\} / E\left\{\left|Y_t^{(.)}(\alpha)\right|^2\right\} \quad \text{(III.12)}$$

Et ensuite, on prend la partie réel de :$\rho^{(.)}(\alpha).e^{-j\theta}$, pour obtenir le signal image.

$$Y_\theta^{2(.)}(\eta_k) = \Re\{\rho(\alpha).e^{-j\theta}\} \quad \text{(III.13)}$$

III.4.3 Bloc de mesure des distorsions

La détection de contour basée sur une version discrétisée. de la mesure de distorsion $\widehat{K_\Omega}$ donnée dans la relation (III.7).

Plus spécifiquement, la distorsion en un point **'t',** notée par $\widehat{K_{\Omega t}}$**,** est calculée avec θ fixe et m spécifie l'espace de η_k dans$[\eta_\alpha, \eta_b]$. (*Comme on a indiqué dans le bloc du filtre paramétrique*). Et, on mesure ces distorsions par deux techniques différentes :

a) La première s'appelle : la divergence spectrale (**KL**) , notée : « la méthode **D_{KL}** ».

b) La deuxième dite : la distance spectrale (**LL2**) notée : « la méthode **D_{LL2}** ».

- ***Méthode DKL***

En utilisant θ fixe et bien précise avec la longueur du segment N variable, la trajectoire de $K_{\Omega t}$ est calculée comme suit :

$$\widehat{K_{\Omega,t}} = \sqrt{N}.\frac{1}{m+1}\sum_{k=0}^{m}\left(K\left[\frac{P_{\theta,t,k}^{(1)}}{P_{\theta,t,k}^{(2)}}\right] + K\left[\frac{P_{\theta,t,k}^{(2)}}{P_{\theta,t,k}^{(1)}}\right]\right) \quad \text{(III.14)}$$

Où :

$P_{\theta,t,k}^{(.)} = \gamma_{\theta,t}^{(1)}(\eta_1) + 1 \quad pour\ K = 0$;

ET :

$P_{\theta,t,m}^{(.)} = 1 - \gamma_{\theta,t}^{(.)}(\eta_m) \quad pour\ K = m$;

ET:

$P_{\theta,t,m}^{(.)} = \gamma_{\theta,t}^{(.)}(\eta_{k+1}) - \gamma_{\theta,t}^{(.)}(\eta_k) \quad pour\ K = 1, \ldots\ldots, m_{-1}$;

- ***Méthode DLL2***

Cette méthode est basée sur les distorsions $\gamma_{\theta,t}^2$ qui sont données par l'expression suivante:

$$\gamma_{\theta,t}^2 = \sqrt{N}\frac{1}{m}\sum_{k=1}^{m}\left|\gamma_{\theta,t}^{(1)}(\eta_k) - \gamma_{\theta,t}^{(2)}(\eta_k)\right|^2 \qquad \text{(III.15)}$$

Avec : θ fixe de façon précise et $\gamma_{\theta,t}^{(1)}(\eta_k), \gamma_{\theta,t}^{(2)}(\eta_k)$ sont les deux signaux filtrés résultats du filtre paramétrique.

III.5 Organigramme de la technique FP

La segmentation d'images à base de la méthode FP se résume par l'organigramme de la figure (III.3) suivante:

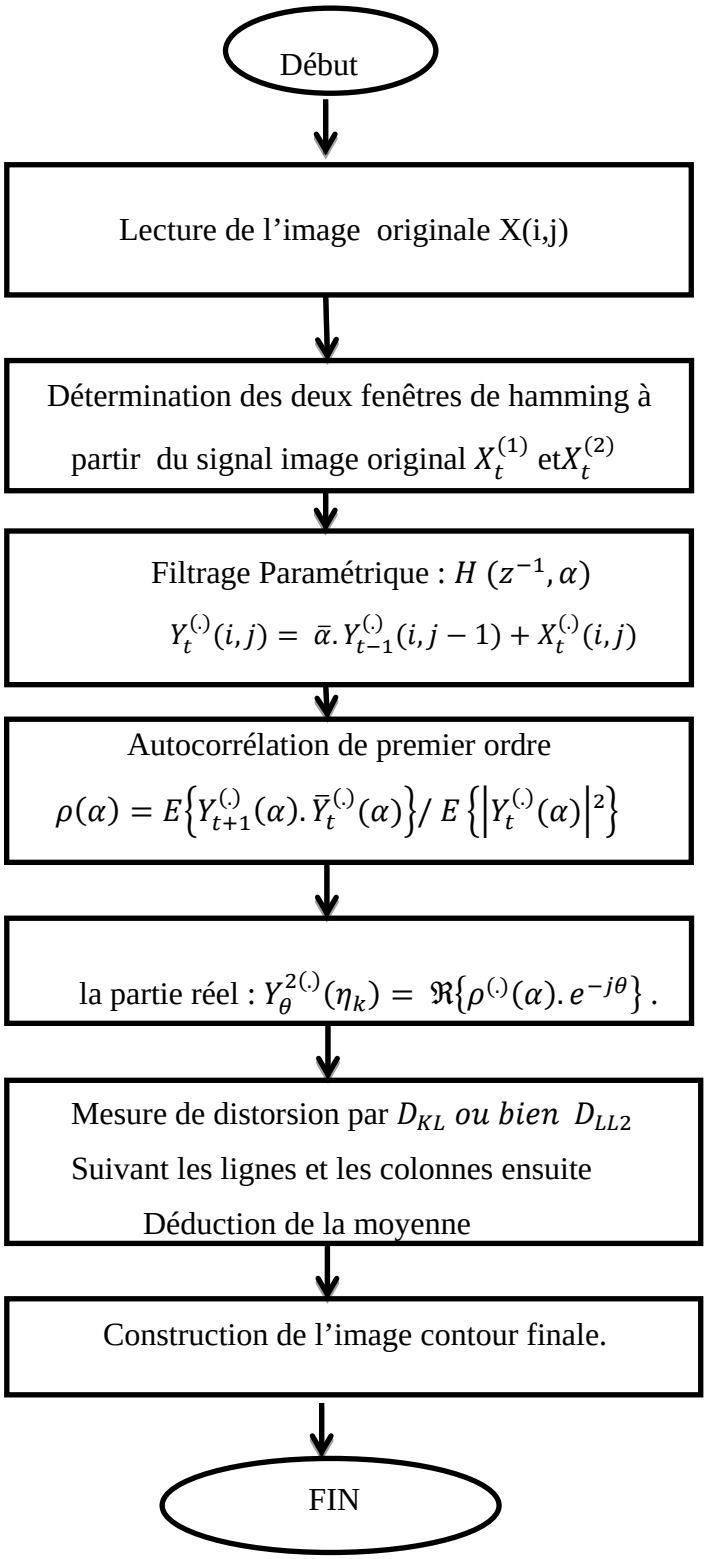


Figure III.3 *L'organigramme de segmentation d'images par la méthode (FP).*

III.6 Implémentation de la méthode FP

Dans cette partie nous allons mettre en œuvre la méthode FP, en utilisant le logiciel MATLAB comme outil de programmation, pour la détection des contours des images, car on procède au suivant :

- Dans un premier temps, on va valider notre méthode pour différent tailles de segmentation N= 2 ; 4 ; 8 ; 16, et pour des valeurs variables de phase du filtre, et ça pour la fenêtre : hamming ; afin de déduire les paramètres qui permettent de bons résultats. Notons que :

- Dans un deuxième temps, on utilise les bons paramètres obtenus précédemment pour appliquer la méthode FP à la segmentation des différents types d'images en faisant une comparaison entre les résultats obtenues pour la DKL et la DLL2

III.6.1 Performance de la méthode FP sans seuillage

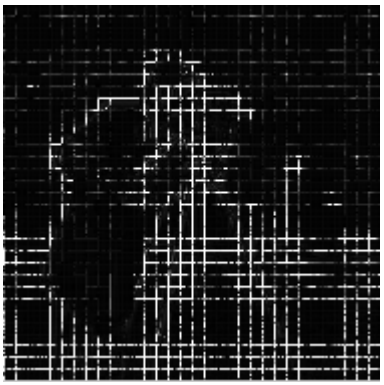
Pour voir l'effet des paramètres sur la détection, nous avons procéder à l'exécution de la méthode sous les paramètres suivantes :

- Image de taille [LxC], avec L et C sont des tailles puissance de deux (2).
- La longueur du segment N prend des valeurs [2 ; 4 ; 6 ; 8 ; 16].
- η_a ; η_b qui prennent des valeurs fixes (bornes d'intervalle [η_a ; η_b]).
- Spécification du nombre d'éléments m dans l'intervalle [η_a ; η_b].
- Choix des valeurs variable de la phase du filtre θ dans le domaine [-π; π].

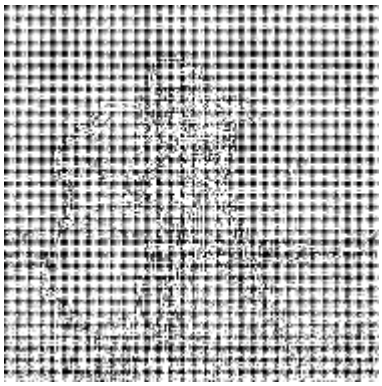
Pour cette étude, nous avons utilisé la fenêtre de hamming pour une image NB (cameraman.jpg) de taille 256x256 où les résultats obtenus pour les différents cas sont les suivants:



Figure III.4.*Image originale.*

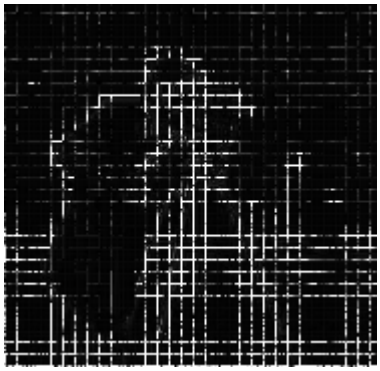


(a).méthode D_{LL2}

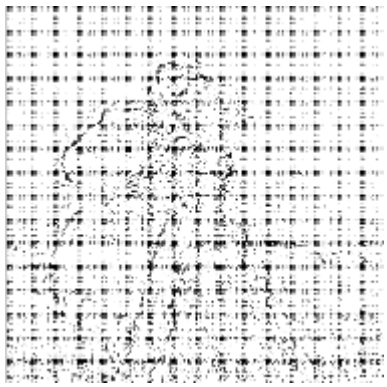


(b).méthode D_{kl}

Figure III.5. *Détection de contour par la méthode PF pour :* $\eta_a = -0{,}3$ *,:* $\eta_b = 0{,}3$ *, N=8, m=4,* θ *= (-pi).*

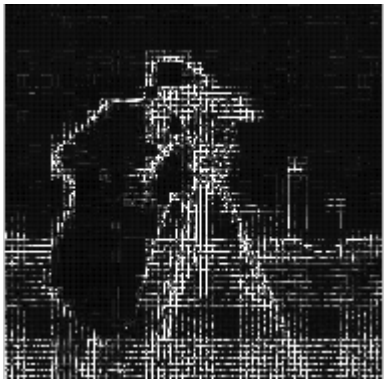


(a).méthode D_{LL2}



(b).méthode D_{kl}

Figure III.6.*Détection de contour par la méthode PF pour :* $\eta_a = -0{,}3$ *,:* $\eta_b = 0{,}3$ *, N=16, m=4,* θ *= (-pi).*



(a).méthode D_{LL2}

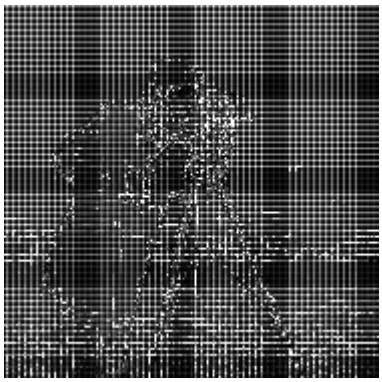


(b).méthode D_{kl}

Figure III.7. *Détection de contour par la méthode PF pour :* $\eta_a = -0{,}3$ *,:* $\eta_b = 0{,}3$ *, N=4, m =2,* θ *= (-pi).*

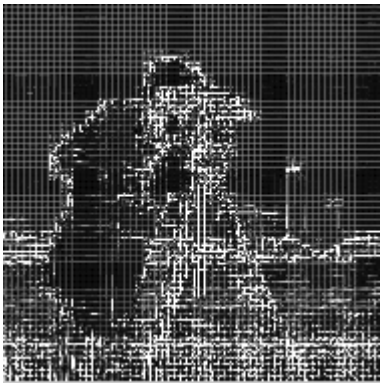


(a).méthode D_{LL2}

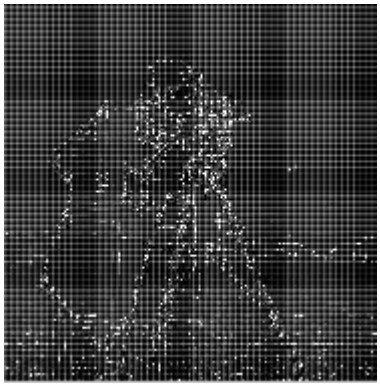


(b).méthode D_{kl}

Figure III.8. *Détection de contour par la méthode PF pour :* $\eta_a = -0.9$ *,:* $\eta_b = 0.9$ *, N=4, m=2,* θ *= (-pi).*

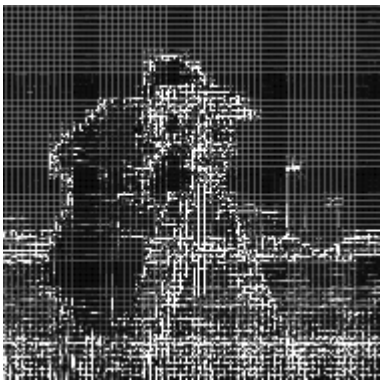


(a).méthode D_{LL2}

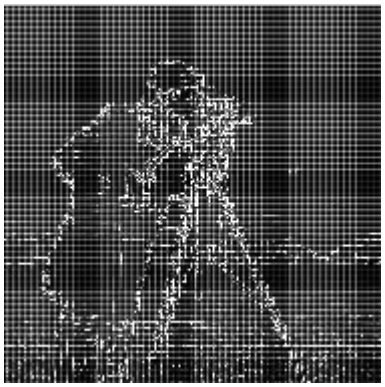


(b).méthode D_{kl}

Figure III.9. *Détection de contour par la méthode PF pour :* $\eta_a = -0.9$ *,:* $\eta_b = 0.9$ *, N=4, m=2,* θ *= pi.*



(a).méthode D_{LL2}



(b).méthode D_{kl}

Figure III.10. *Détection de contour par la méthode PF pour :* $\eta_a = -0.9$ *,:* $\eta_b = 0.9$ *, N=4, m=2,* θ *= 0.025*pi.*

Commentaire

D'après les résultats obtenus pour la détection de contour sans seuillage, on constate que les deux méthodes de détection de contour D_{KL} et D_{LL2} donnent de bons résultats sous certaines paramètres bien précises et qui sont: $\eta_a = -0.3$,: $\eta_b = 0.3$, N=4, m=2, θ= (-pi).

D'autre part, la méthode D_{LL2} donne aussi de bons résultats pour d'autres paramètres comme il est montré sur les figures précédentes.

III.6.2 Performance de la méthode FP avec seuillage

Pour qu'on puisse déterminer les contours avec des meilleures caractéristiques, dont on va utiliser la détection de contour avec seuillage sous les paramètres optimales suivants :

- Pour la méthode D_{LL2}, on utilise les paramètres suivant :

  $\eta_a = -0{,}3$, $\eta_b = 0{,}3$, N=4, m =2, θ= (-pi).

- Pour la méthode D_{KL} on utilise les paramètres suivant :

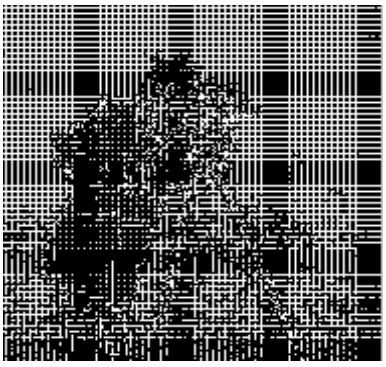
  $\eta_a = -0.9$,: $\eta_b = 0.9$, N=4, m=2, θ= 0.025*pi.

L'intégration du seuillage consiste à effectuer une opération de limitation des puissances du signal contour obtenu par les DKL (ou la DLL2). D'où on distingue le seuillage normal et la façon de travailler dans ce genre consiste à déterminer le niveau de seuil et, par conséquent de la frontière, ce qui est inférieur au niveau de seuil sont une prise, la majeure partie au niveau de seuil est pris 0. En ce qui concerne la deuxième méthode de seuillage par hystérésis, son chemin va dépendre du choix d'une zone spécifique du niveau de seuil, à partir des seuils max et min. en appliquant ces méthodes sur une image (cameraman.jpg) de taille (256x256).

a- Résultat de détection de contour avec seuillage normale



(a).méthode D_{LL2}



(b).méthode D_{kl}

Figure III.11. *Détection de contour par la méthode DKL et la méthode DLL2 avec seuillage normale.*

b- Résultat de détection de contour avec seuillage hystérésis



(a).méthode D_{LL2} (b).méthode D_{kl}

Figure III.12. *Détection de contour par la méthode DKL et la méthode DLL2 avec seuillage hystérésis.*

Commentaire

D'après les résultats obtenus, on constate que les deux méthodes de détection de contour DKL etDLL2 donnent de bons résultats sous les paramètres précédemment déterminés et avec les niveaux de seuil suivant :

- pour la méthode DLL2 pour les deux cas du seuillages:

(**Seuil normal = 0.25 ; seuil hystérésis haut = 0.35 ; seuil hystérésis bas = 0.25).**

- pour la méthode DKL pour les deux cas du seuillages :

(**Seuil normal = 0.7 ; seuil hystérésis haut = 0.35 ; seuil hystérésis bas = 0.25).**

III.7 Performance de détection en fonction des diverses fenêtres

Dans cette partie nous utilisons les meilleurs paramètres pour D_{KL} et D_{LL2} qui sont : $\eta_a = -0.3$,: $\eta_b = 0.3$,N=4, m=2, θ= (-pi), pour les différentes fenêtres (Hamming, Hanning, Blackman, Chapeau mexicain, Meyer, Morlet).

III.7.1 Pour la méthode DLL2 :

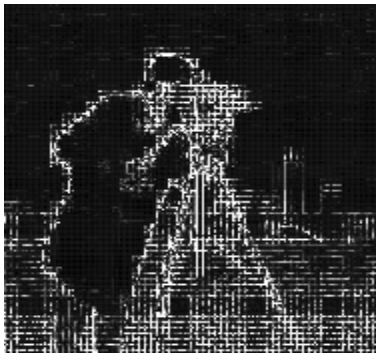
fenêtre Hamming	fenêtre Hanning	fenêtre Blackman
		

Figure III.13. *Détection de contour par la méthode DLL2 par l'utilisation de la fenêtre de Hamming ; Hanning ; blackman.*

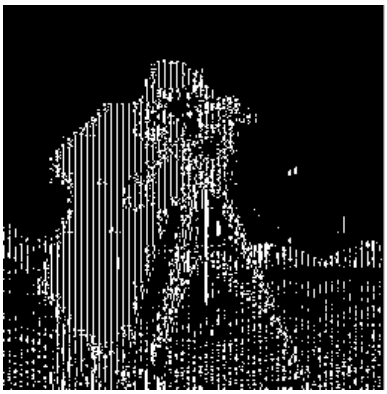
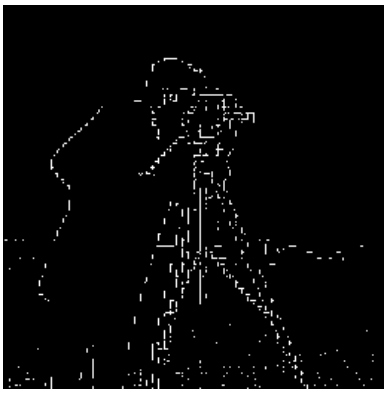
Ondelette Chapeau mexicain	Ondelette Meyer	Ondelette Morlet
		

Figure III.14. *Détection de contour par la méthode DLL2 par l'utilisation de l'ondelette Chapeau mexicain ; ondelette Meyer; ondelette Morlet.*

III.7.2 Pour la méthode D_{kl}

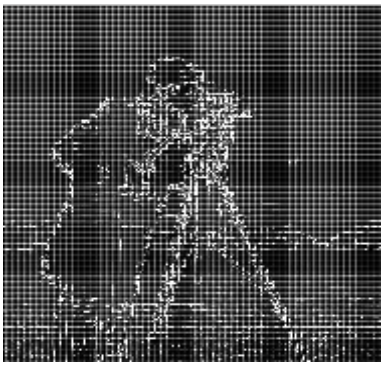
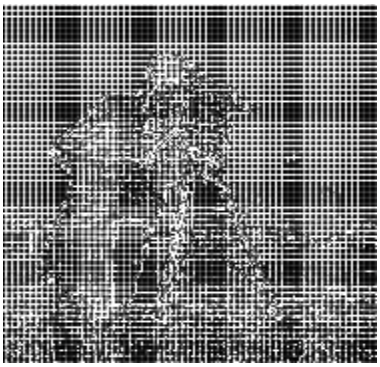
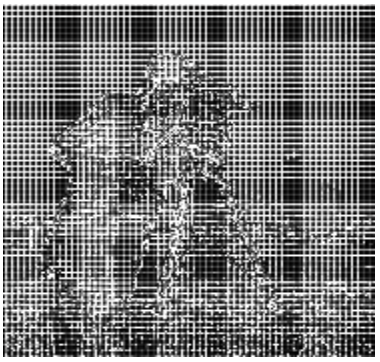
fenêtre Hamming	fenêtre Hanning	fenêtre Blachman
		

Figure III.15. *Détection de contour par la méthode DKL par l'utilisation de la fenêtre de Hamming, Hanning , blackman .*

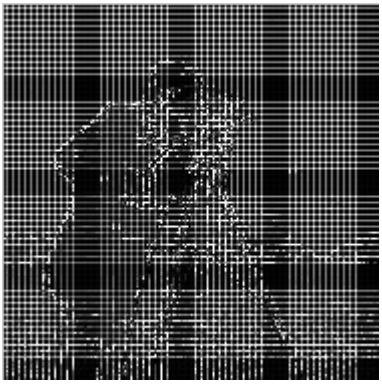
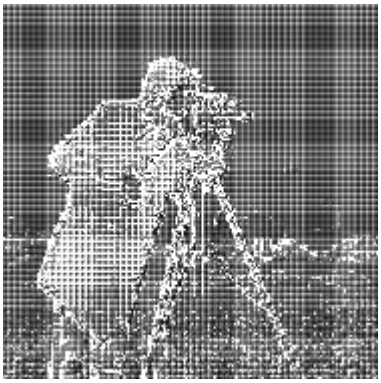
Ondelette Chapeau mexicain	Ondelette Meyer	Ondelette Morlet
		

Figure III.16.*Détection de contour par la méthode DKL par l'utilisation de l'ondelette Chapeau mexicain ; ondelette Meyer; ondelette Morlet.*

Commentaire

D'après les résultats obtenus, on constate que l'utilisation de la fenêtre de Hamming permet la détection des contours d'une manière plus fine par rapport à celles des autres fenêtres (Hanning et Blackman).

Donc, en ce qui suit on tient compte de l'utilisation de la fenêtre de Hamming.

III.8 Comparaison entre DKL et DLL2

Nous remarquons, d'après les résultats précédents que la méthode de filtrage paramétrique donne de résultat relativement bons coté contour, l'image résultats ne contient pratiquement pas de bruit et les chaînes de contour sont continués et cela selon le choix des paramètres. Ceci est dû grâce à l'utilisation du passage par zéro de la différence des pics de la trajectoire de la distorsion et surtout grâce au type du filtre qui est à réponse impulsionnelle infinie (récursive et séparable)…

D'après les résultats de détection des images seuillées, nous constatons aussi que la méthode paramétrique (D_{LL2}) est très robuste contre les petites et ‹‹insignifiantes›› variations dans le domaine spectrale de l'image.

Coté temps de calcul, l'algorithme de la méthode paramétrique D_{LL2} est meilleure par rapport au D_{KL} est très couteux en temps de calcul, mais il n'est pas nécessaire pour certains domaines d'application (médical par exemple,…), d'avoir un algorithme rapide.

Nous constatons aussi que le filtrage paramétrique détecte les contours même ceux ayant une faible intensité lumineuse.

Pour les processeurs de lissage et de détection en même temps, c'est la méthode paramétrique (D_{LL2}) qui présente de bonne détection et bonne localisation si on applique le seuillage par suppression des points non maximums ; on remarque que tous les détails essentiels sont visibles, bien que certains points parasites subsistent à l'intérieur des objets, on arrive à avoir des contours continus.

Cette méthode permet de généraliser les techniques de détection de contours en considérant, en cascade un filtre passe bande (récursive et séparable) pour lissage et un opérateur de corrélation qui détecte les points maximums, ceux-ci regroupés pour former des chaînes de contours.

III.9 Conclusion

Dans ce chapitre, une nouvelle méthode de détection de contour d'image et de caractérisation de l'image est présentée. La méthode PF emploie un filtre judicieusement déterminé, qui préserve la structure de corrélation du signal d'entrée pour obtenir l'autocorrélation $\gamma_\theta(\eta)$ à la sortie. Ceci amène bien l'évolution spatiale de la structure de corrélation de l'image que diverses mesures de distorsion qui quantifient la déviation entre deux zones du signal (*Les deux signaux obtenus de Hamming*) pour la détection de contour d'une image.

Aussi nous avons présenté les différents résultats obtenus par notre méthode PF. Pour ce fait, on a testé les performances de notre méthode, en faisant, en variant les différents paramètres en présence des différentes fenêtres (Hamming, Hanning, Blackman, ondelette Chapeau mexicain, ondelette Meyer et ondelette Morlet) pour spécifier les paramètres optimaux qui permettent de bons résultats.

Nous avons trouvé, d'après les résultats obtenus que l'efficacité de notre méthode de filtrage paramétrique n'est possible que pour des paramètresη_a , η_b, N, m, et θ et aussi suivant les paramètres d'ondelettes spécifiés.

Pour remédier à ce problème de temps lourd de traitement, nous allons proposer une méthode qui nommée détection de contour d'image par bloc.

CHAPITRE IV

Extension de la méthode PF en détection de contour d'image fixe par blocs

IV.1 Introduction

Nous allons présenter dans ce chapitre une amélioration de la méthode étudiée au chapitre précédent de point de vue de performances de la qualité de contours obtenus et diminution de temps d'exécutions.

Cette extension base sur le principe de décomposition de l'image fixe originale en blocs de petits dimensions afin d'assurer la connexité des pixels du bloc.

Notre approche base sur l'application de la technique PF pour chaque bloc suivant ses lignes et ses colonnes par divers types de fenêtre ; première cas en utilisant la fenêtre Hamming et pour le deuxième cas on va utiliser les divers types d'ondelettes.

Pour l'implémentation des deux algorithmes, nous avons utilisé le Logiciel Matlab 7.8.0 R2009a.

IV.2Principe

Le schéma suivant montre la procédure de décomposition de l'image en blocs de de tailles MxM. :

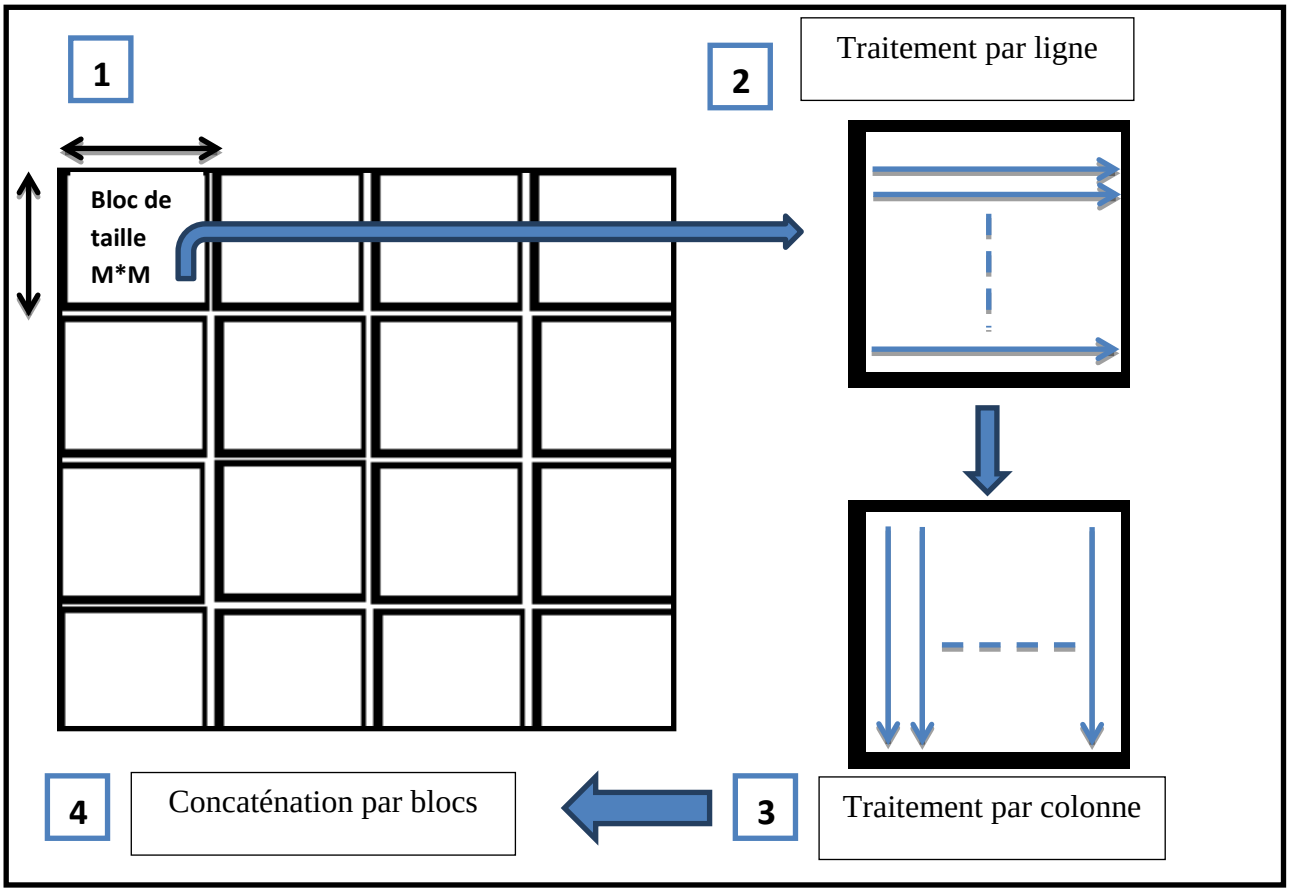


Figure IV.1 *Schémas explicatifs de la méthode de détection de contours par bloc*

IV.3 Organigramme de détection de contour d'image par blocs

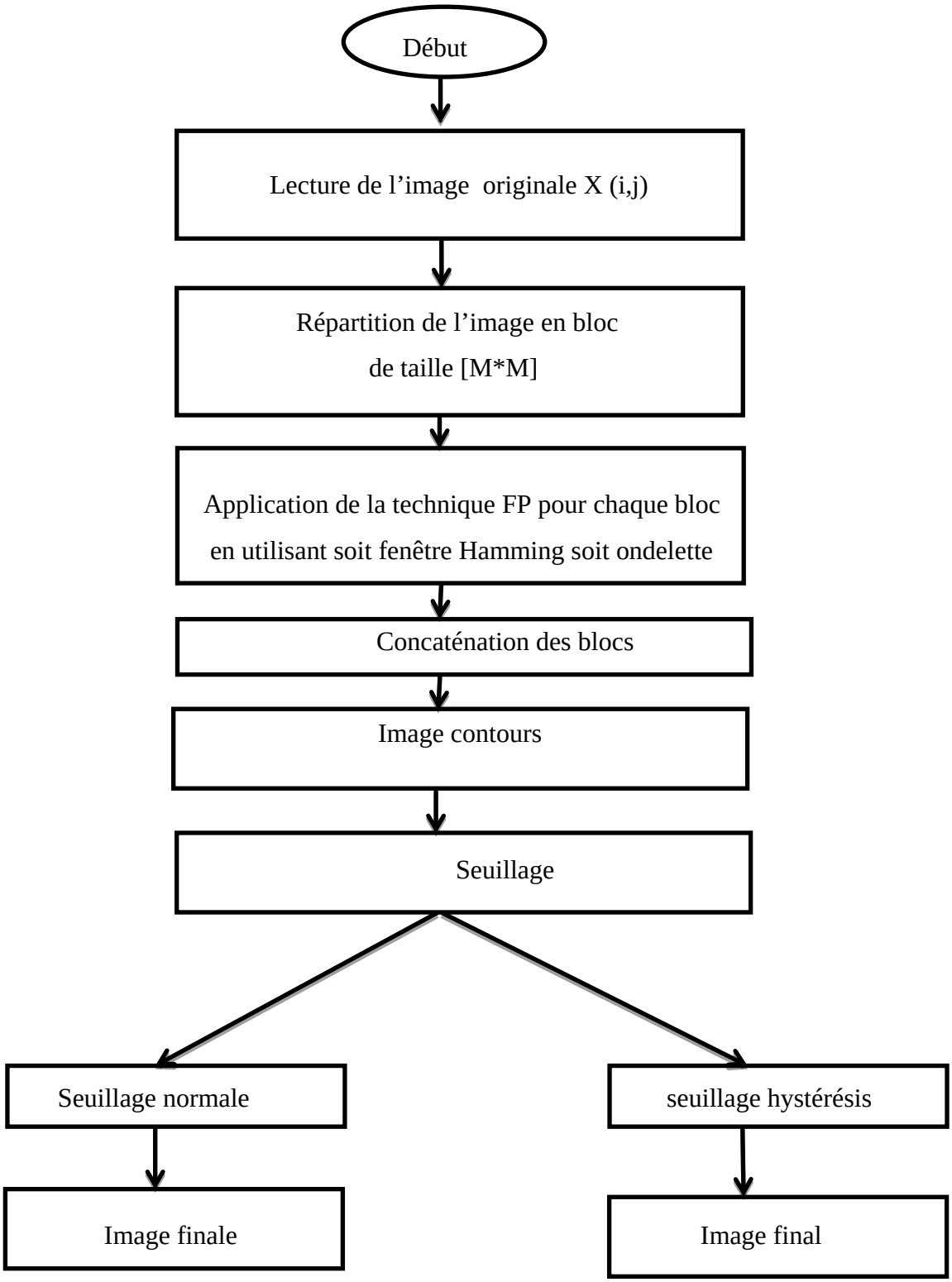


Figure IV.2 *L'organigramme de détection de contour d'image par blocs*

IV.4 Mise en œuvre et discussion des résultats

IV.4.1 premier cas

Dans ce qui suit, on présente les résultats de détection de contour par la première méthode (fenêtre hamming) en prenant les meilleurs meilleures caractéristiques du filtrage paramétrique du chapitre précèdent pour comparer les résultats obtenus avec la méthode précédente l' image (wpeppers.jpg), ensuite on joue sur les tailles du bloc pour dire ce qui est la meilleur taille de bloc, et finalement on applique cet algorithme sur une image couleur.

À la fin on va discuter le résultat.

- **Pour la méthode D_{LL2}, on utilise les paramètres suivant :**

  $\eta_a = -0{,}5$, $\eta_b = 0{,}5$, N=4, m =2, θ= (-pi).

  (Seuil normal = 0.25 ; seuil hystérésis haut = 0.35 ; seuil hystérésis bas = 0.25).

- **Pour la méthode D_{KL} on utilise les paramètres suivant :**

  $\eta_a = -0.9$,: $\eta_b = 0.9$, N=4, m=2, θ= 0.025*pi.

  (Seuil normal = 0.7 ; seuil hystérésis haut = 0.35 ; seuil hystérésis bas = 0.25).

❖ ***Détection des contours des images en niveau de gris***

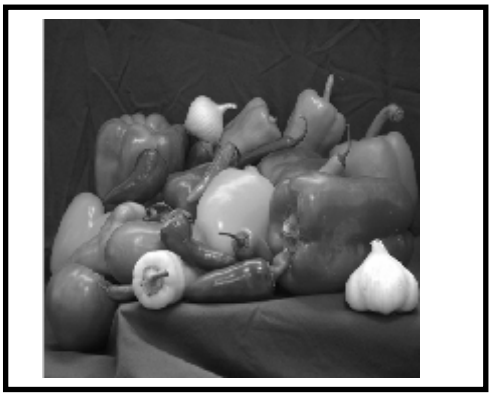
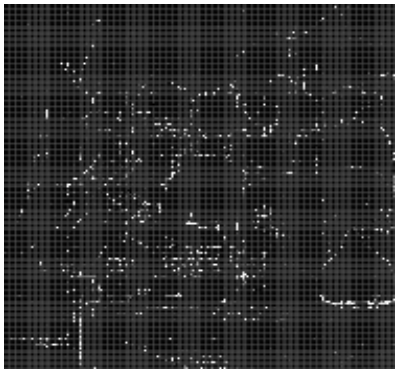
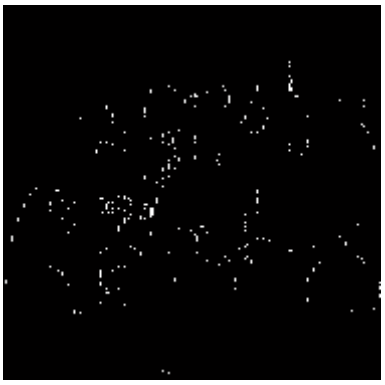


Figure IV.3.*Image originale.*

a) la méthode FP par bloc sans seuillage



(a).méthode D_{LL2}



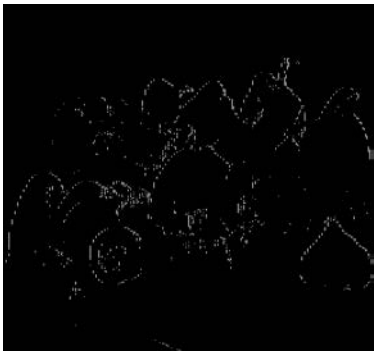
(b).méthode D_{kl}

b) la méthode FP par bloc après seuillage

- *seuillage normale :*



(a).méthode D_{LL2}



(b).méthode D_{kl}

- *seuillage hystérésis :*



(a).méthode D_{LL2}



(b).méthode D_{kl}

Commentaire

Apres plusieurs essais et observations, on constate que la méthode de détections par blocs donne des meilleurs résultats par rapport la méthode FP classique.

En plus on remarque que le seuillage normal est meilleur que le seuillage hystérésis sous les paramètres bien précis qui ont déclarés au-dessus.

D'autre part on trouve toujours que la méthode DLL2 donne des résultats plus précis par rapport la méthode DKL.

❖ **Détection des contours d'une image couleur en différentes tailles de bloc**

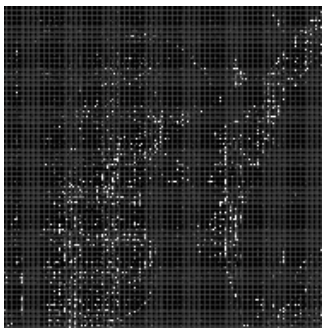


Figure IV.4.*Image originale.*

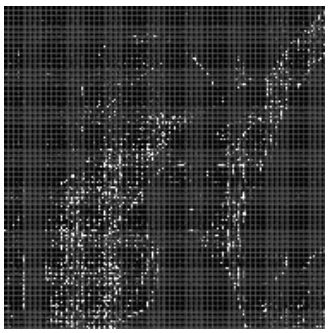
a) la méthode FP par bloc sans seuillage



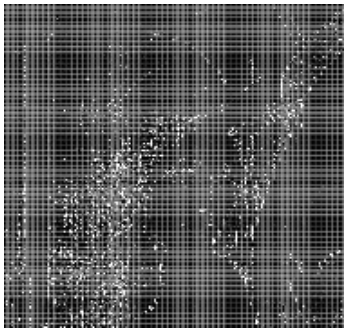
Bloc= 64 methode dkl



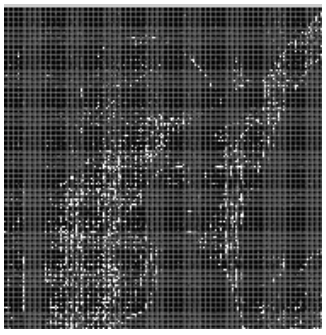
Bloc= 32 methode dkl



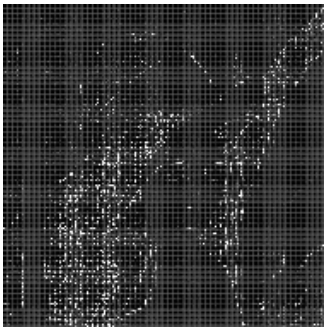
Bloc= 4 methode dkl



Bloc= 64 methode DLL2



Bloc= 32 methode DLL2



Bloc= 4 methode DLL2

b) la méthode FP par bloc après seuillage

❖ *Seuillage normale :*



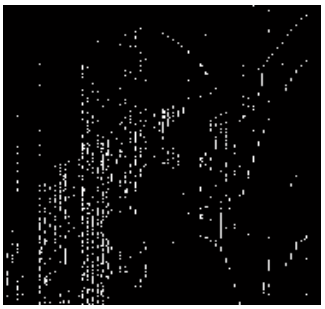
Bloc= 64 methode dkl



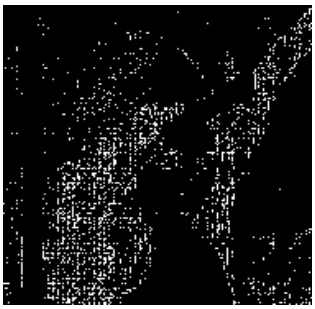
Bloc= 32 methode dkl



Bloc= 4 methode dkl



Bloc= 64 methode DLL2



Bloc= 32 methode DLL2

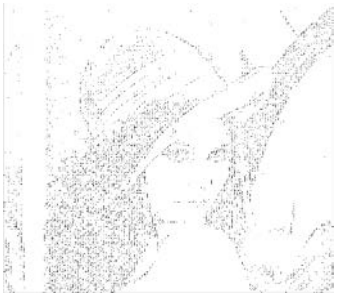


Bloc= 4 methode DLL2

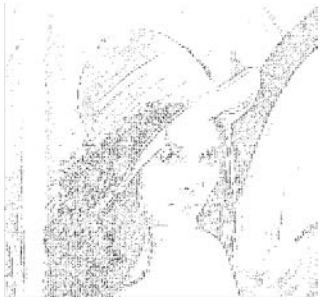
- *seuillage hystérésis :*



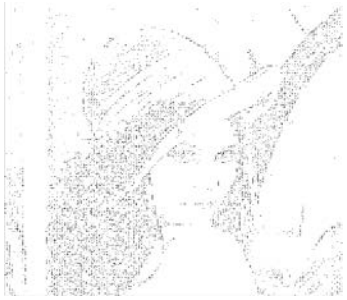
Bloc= 64 methode dkl



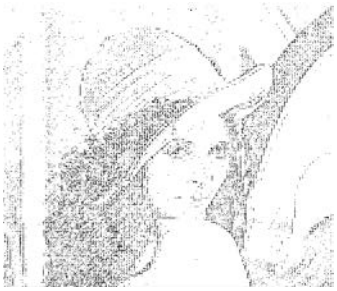
Bloc= 32 methode dkl



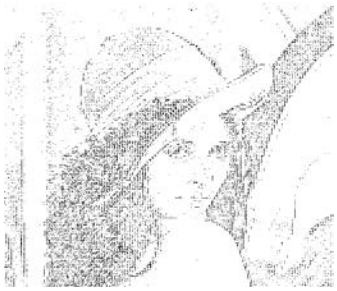
Bloc= 4 methode dkl



Bloc= 64 methode DLL2



Bloc= 32 methode DLL2



Bloc= 4 methode DLL2

Commentaire

D'après les résultats obtenus nous pouvons dire que la méthode FP de détection de contours par blocs à apporter une bonne amélioration surtout de point de vue qualité de détection dans les deux procédures (D_{KL}, D_{LL2}).

D'une part nous remarquons que la qualité de détection à une relation inversement proportionnelle avec la taille des blocs choisis; si la taille est grande la qualité de détection devient médiocre et le contraire et juste, si la taille était petites la qualité de contours sera bonne mais le temps de l'exécution sera lent.

D'autre part la détection des contours des images N &B est très rapides par rapport à celles des images couleurs et à niveau de gris puisque le traitement se fera sur une seul matrice représentative des données.

IV.4.2Deuxième cas

Dans cette implémentation une succession d'essaies nous permets le choix des paramètres convenables pour différentes familles d'ondelettes. D'autre part, l'image (persan.jpg) est prise comme images fixe sur laquelle on applique la détection de contour.



Figure IV.5.*Image originale.*

A- Détection de contour par la méthode FP par blocs pour une Ondelette ‹‹Chapeau Mexicain››

- **Pour la méthode DLL2**



Sans seuillage seuillage normale seuillage hystérésis

Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

$\eta_a = -0{,}3$, $\eta_b = 0{,}3$, N=4, m =2, θ= (0.0025*pi).

(Seuil normal = 0.5 ; seuil hystérésis haut = 0.7 ; seuil hystérésis bas = 0.5).

- **Pour la méthode DKL**



Sans seuillage seuillage normale seuillage hystérésis

Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

$\eta_a = -0{,}5$, $\eta_b = 0{,}5$, N=4, m =2, θ= pi.

(Seuil normal = 0.35 ; seuil hystérésis haut = 0.6 ; seuil hystérésis bas = 0.35).

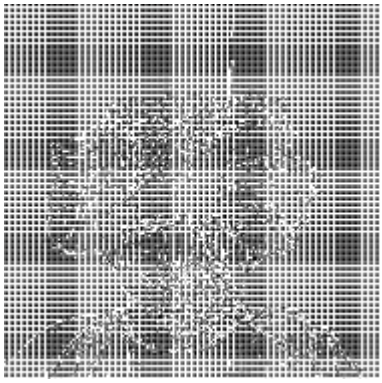
Commentaire

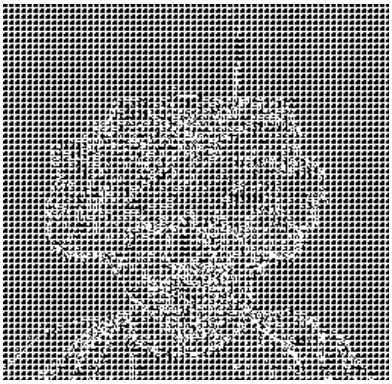
En utilisant l'ondelette chapeau mexicain, on constate qu'elle nous donne dans les deux méthodes de détection D_{KL} et D_{LL2} de bons résultats sous certaines conditions bien précise.

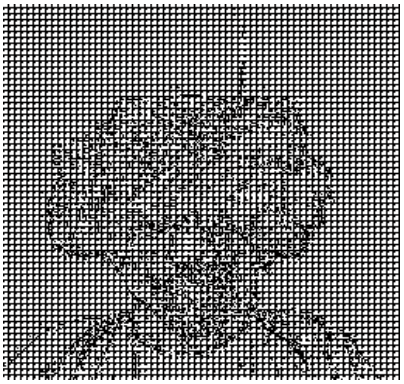
En plus on trouve que le seuillage normal est meilleur que le seuillage hystérésis sous certains paramètres bien précis après de nombreux essais.

B- Détection de contour par la méthode FP par blocs pour une Ondelette ‹‹Meyer››

- **Pour la méthode DLL2**


Sans seuillage


seuillage normale

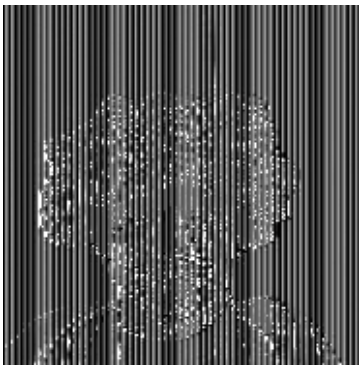

seuillage hystérésis

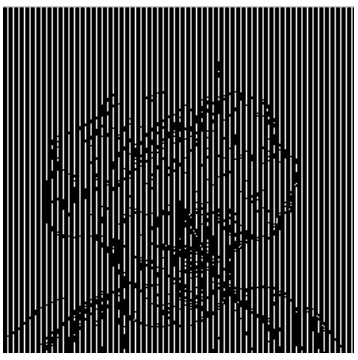
Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

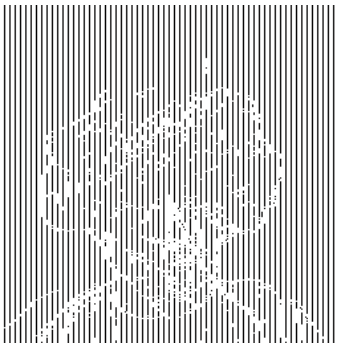
$\eta_a = -0{,}2$, $\eta_b = 0{,}2$, N=4, m =2, θ= pi.

(Seuil normal = 0.4 ; seuil hystérésis haut = 0.65 ; seuil hystérésis bas = 0.45).

- **Pour la méthode DKL**


Sans seuillage


seuillage normale


seuillage hystérésis

Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

$\eta_a = -0{,}45$, $\eta_b = 0{,}45$, N=8, m =4, θ= -pi.

(Seuil normal = 0.4 ; seuil hystérésis haut = 0.65 ; seuil hystérésis bas = 0.45).

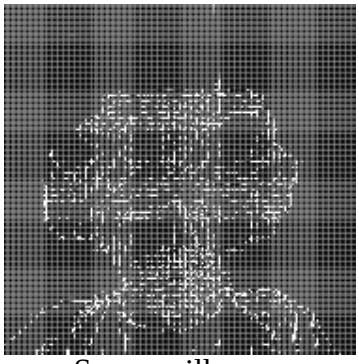
Commentaire

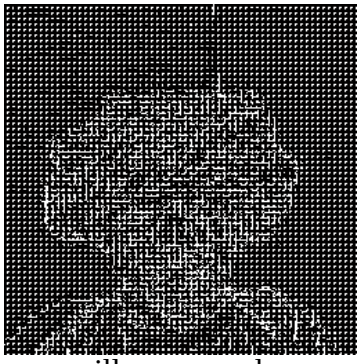
En utilisant l'ondelette Meyer, on constate qu'elle nous donne pas une bonne détection dans les deux méthodes D_{KL} et D_{LL2} par rapport à l'ondelette précédente.

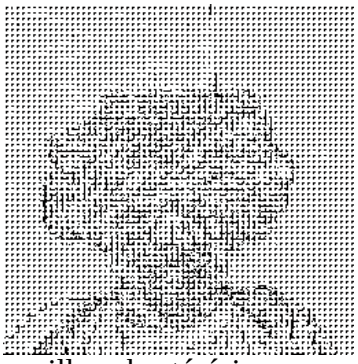
En plus on trouve que le seuillage normal est meilleur que le seuillage hystérésis sous certains paramètres bien précis après de nombreux essais.

C- Détection de contour par la méthode FP par blocs pour une Ondelette ‹‹Morlet››

- **Pour la méthode DLL2**


Sans seuillage


seuillage normale


seuillage hystérésis

Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

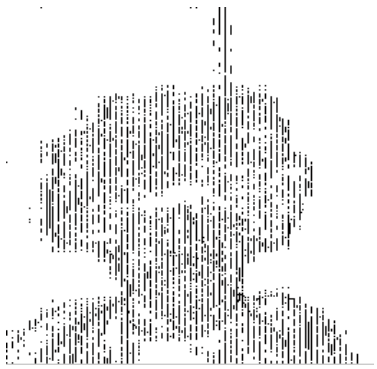
$\eta_a = -0{,}2$, $\eta_b = 0{,}2$, N=4, m =2, θ= pi.

(Seuil normal = 0.65 ; seuil hystérésis haut = 0.55 ; seuil hystérésis bas = 0.35).

- **Pour la méthode DKL**


Sans seuillage


seuillage normale


seuillage hystérésis

Les paramètres utilisés pour la détection de contour après beaucoup d'essais sont :

$\eta_a = -0{,}2$, $\eta_b = 0{,}2$, N=4, m =2, θ= pi.

(Seuil normal = 0.65 ; seuil hystérésis haut = 0.55 ; seuil hystérésis bas = 0.35).

Commentaire

En utilisant l'ondelette Morlet, on constate que la qualité de détection de cette ondelette est au milieu (entre la qualité de l'ondelette chapeau mexicain et l'ondelette Mayer) dans les deux méthodes D_{KL} et D_{LL2}.

En plus on trouve que le seuillage normal est meilleur que le seuillage hystérésis sous certains paramètres bien précis après de nombreux essais.

Remarque

Donc l'ondelette Chapeau mexicain est bien plus meilleure que les autres ondelettes.

IV.5Conclusion

Dans ce chapitre nous avons effectuées différentes manipulations sur une image fixe noir et blanc et l'autre en couleur qui ont montres l'efficacité de la méthode de détection de contours par blocs.

L'utilisation des déférentes ondelettes a montré aussi une grande amélioration dans la détection d'où nous avons trouvé la visualisation des contours fins.

Nous avons trouvé que, sous des paramètres soigneusement choisis et des niveaux de seuillages adaptatifs, les deux techniques DKL et DLL2 présentent de bons résultats pour différents familles d'ondelettes utilisés avec des coefficients de dilatation précis, malgré la difficulté du choix des paramètres de chaque type d'ondelettes. Donc on peut dire que la méthode FP à base d'ondelettes est meilleur comparée à celle à base de la fenêtre de Hamming.

Conclusion générale

Ce travail est entrepris dans le but de déterminer l'effet de la méthode de détection des contours sur la qualité du traitement des images numériques, On faisant une étude des performances de détection suivant des divers paramètres tel que la longueur de segment choisi ainsi que la phase et le niveau de seuillage pour chaque type de fenêtre (Haming, Hanning, Blachman), et divers famille d'ondelettes (Chapeau mexicain, Meyer, Morlet), d'autre part nous avons proposé une approche de détection par bloc afin d'améliorer les performances de détection et de démineur le temps d'excusions.

En utilisant, les fenêtres Hamming, Hanning blackman, les résultats obtenus par la fenêtre de Hamming sont meilleurs.

Dans le cas des types d'ondelettes (chapeau mexicain, Haar et Morlet), les résultats associés au Chapeau mexicain sont meilleurs.

La qualité de détection des contours est inversement proportionnelle avec la taille des blocs.

Le filtrage paramétrique par ondelettes donne des performances meilleures de détection par rapport au filtrage par fenêtre de Hamming.

La vitesse de traitement est inversement proportionnelle avec la taille des blocs.

La méthode DLL2 donne des qualités meilleures par rapport à celle de la méthode DKL.

Il est suggéré de choisir le seuillage normal pendant le filtrage paramétrique.

Comme perspective nous proposons une étude de détection de contours par une approche amélioré PF en 2D .

Bibliographie

[01] R.C. GONZAILES & P. WINTZ. “Digital Image Processing”, Addison Wessley, 1977.

[02] WILLIAM PUECH ; ’’ Vision par ordinateur’’ ; Université Montpellier II- Nîmes.

[3] J. ANTOINE et L. JACQUES, ‘’Traitement de l’image’’, Université Catholique de Louvain;

[4] A.MANZANERA ‘’Traitement et reconnaissance d'images’’ ,*Cours TERI – Master IAD UPMC Paris 6.*

[05] Microsoft Encyclopédie ENCARTA, 1997.

[06] L.MOHAMMEDI, M.MELIANI ; ‘’étude comparative de différents algorithmes de détection de contours’’ ; PFE Blida ; 1994.

[07] K. TABARI & S. TAGMH. ‘’Compression d’images animées à très débit par la géométrie des fractales’’. PFE USTHB 1996.

[08] Y.KABIR, S.DARBANE. ‘’étude et implémentation des méthodes déterministes de segmentation d’image par extraction de contours’’. PFE Blida,1995.

[09] J-P. COCQUEREZ & S. PHILLIP. ‘’Analyse d’image, filtrage et segmentation’’. Edition Masson, Paris, 1995.

[10] ANTOINE MANZANERA. ‘’Traitement d'images et vision artificielle’’. ENSTA D9-2

[11] M.HADAL-LAH ;’’ Codage des images fixes par une méthode hybride basée sur la QV et les approximations fractales’’ ; PFE USTHB ; 1997.

[12] GUY MELARD. ‘’Initiation à l’analyse des séries temporelles et à la prévision de RevueMODULAD, 2006.
[13] G.M. COCHARD, S.CHOPLIN ;’’ infographie et image informatique’’.

[14] LAKHDARI MOHAMED. ‘’Segmentation d’images par contour actif en appliquant les algorithmes génétiques’’. PFE Alger, 2008.

[15] E. BOURENNANE, M. PAINDAVOINE, F TRUCHETET. ‘’Amélioration du filtre CANNY-DERICHE pour la détection des contours sous forme de rampe’’. Traitement du signal, vol. 10, N0 4,pp. 297-310,1993.

[16] R. ADRE-OBRECH. ‘’A new statistical approaches for automatic segmentation of continuous speech signals, IEEE Trans. Acoust., Speech, Signal Processing’’, vol 36.pp 29-40, Jan 1998.

[17] E. VIDAL AND A. MARZAL. ‘’A review and new approach for automatic segmentation of speech signals, in Signal Processing V: Theories and Application’’, L. Torres et al. Eds. New York: Elsevier, 1990, vol 1 pp. 43-53.

[18] A.YAHIA , A.GUESSOUM. ‘’Détection de contour par la méthode de filtrageParamétrique’’ 2005.

[19] T.H. Li AND J.D. GIBSON. ‘’Speech Analysis and Segmentation by parametric filtering IEEE Trans on speech and audio processing’’. Vol.4, No 3 May 1996 pp 203-213.

[20] M. BASSEVILLE. ‘’Distance measures for signal processing and pattern recognition, Signal’’

[21] R.M. GRAY. A. BAZO. Y. MATSUYAMA. ‘’Distortion measures for Speech Processing”, IEEE Trans. Acoust, Speech, Signal Processing’’, vol ASSP- 28, pp. 367-376, Apr. 1980.

[22] J.G. WILPON, B.H. JUAN, AND L.R. RABINER. ‘’An investigation on the use of acoustic sub-word units for automatic speech recognition’’, in Proc. ICASSP, TX, Apr.1987, pp 821-824.

[23] LUC CLAUSTRES,‘’Introduction aux ondelettes’’, 2002-2003.

[24] RENE ALT, ‘’La transformation en ondelettes’’, 2001.

[25] A.COHEN,‘’Techniques de l’Ingénieur Les bases d’ondelettes’’,1999.

[26] Y.BACHEN, A.BENGHARBIA, H.METIDJI, ‘’Comprissions d'image avec perte transformer enOndelette’’, P.S Univ-MEDEA, 2010.

[27] V. PERRIER,‘’Transformée en ondelettes continue théorie, application à l’imagerie médicale’’, 12 décembre 2005.

[28] I.AMARSETTE, O.BAYASLI,‘’Etude et simulation de décomposition en sous-bandes d’image à base d’ondelettes’’ (Quantification et Codage), PFE Univ-MEDEA, 2003.

[29] D.JONATHAN, B.MICHAEL, F.SEBASTIEN,‘’Les Ondelettes’’, 2003.

Programme de filtrage paramétrique DKL :

```
clear all;
clc
format shorte
%-------------------------------------------------------------
%lecture de l'Image originale et initialisation des paramètres
X00 =imread('formes_256.jpg');
X0=rgb2gray(X00);          % transformation d'image en grais
[l,c]=size(X0);            %    dimensions d'image originale
L=l*c;              % nombre total des éléments de la matrice
LS=4;                      %    la longueur du segment;
N=LS;
theta=0.0025*pi;           %    la valeur de la phase teta
na =-0.5;                  % borne inferieur de l'intervalle
nb =-na;                   % borne supérieure  de l'intervalle
m=2;             % nombre d"éléments de l"intervalle [na,nb]

tic

% Preallocate nk pour améliorer les performances Temps
d'execution
nk = zeros(1,m);
%Détermination de Alpha(k) en fonction de néta(k) et theta :

for k=1:m;
    nk(k)=na+((k-1)*(nb-na))/(m-1);
end

Alpha=nk.*exp(1j*theta);
%-------------------------------------------------------------
%***détermination de la fenêttre de hamming symmétrique et
celle décalée:
%
        wh =hamming(LS);
        LS1=fix(LS/2);
        w1(1:LS)=0;
        w1(1:LS)=wh(1:LS      %fenêtre de hamming symétrique.
        w2(1:LS)=0;
        w2(1:LS1)=wh(LS1+1:LS);
        w2(LS1+1:LS)=wh(1:LS1); % fenêtre de hamming décalée.
        n=1;
% x est un vecteur de dimension (1,l*c)
        x(L)=0;
for ii=1:l
            for jj=1:c
                x(n)=X(ii,jj);
      % x: transformation de ma matrice en un vecteur ligne:
                n=n+1;
            end
        end
```

```
%* Segmentation du vecteur Image et les arranger en matrice
segments suivant les lignes:
        % code peut etre amélioré
        % NS= (L/LS)
        % if NS>fix(L/LS)
        %      NS = NS+1;
        %      x(NS*LS)=0;
        % end

        NS=L/LS;
        NS1=fix(NS);
        if NS > NS1
            NS=NS1+1;
            x(NS*LS)=0;
        end
        L1=NS*LS;
        ls=1;
        ns=1;
        % fin code à amélioré
        xls(NS,LS)=0;
        for l1=1:L1
            xls(ns,ls)=x(l1);
            ls=ls+1;
            if ls>LS
                ls=1;
                ns=ns+1;
            end
        end
        xls;

%****Traitement de tous les segments obtenus:
% préallocation
y = zeros(m,LS);
A1 = zeros(m,LS);
y1 = zeros(m,LS);
Y1 = zeros(m,LS);
A2 = zeros(m,LS);
Y2 = zeros(m,LS);
P1 = zeros(m,LS);
P2 = zeros(m,LS);
KK1 = zeros(NS,LS);
        for ns=1:NS
            xss(1:LS)=xls(ns,1:LS);
%
%****** les vecteurs convolutions obtenus dans les deux cas:
%
            xss1=conv(xss,w1);
            xss2=conv(xss,w2);
```

```
%******calcul des paramètres : alpha et neta ainsi que le
filtre suivant la relation yt=alpha1*yt-1+xt:

            for k=1:m;
                y0=1;
                y00=1;
                for ls=1:LS;
y(k,ls)=conj(Alpha(k))*y0+xss1(ls+(LS/2-1));
% filtrage pour le premier signal

A1(k,ls)=mean(y(k,ls)*conj(y0))/mean((abs(y0)).^2);
 % auto correlation 1
y1(k,ls)=conj(Alpha(k))*y00+xss2(ls+(LS/2-1));
% filtrage pour le 2iéme signal
 Y1(k,ls)=real(exp(-1j*theta)*A1(k,ls));
 % autocorrelation modulée

A2(k,ls)=mean(y1(k,ls)*conj(y00))/mean((abs(y00)).^2);
 % autocorrélation 2
Y2(k,ls)=real(exp(-1j*theta)*A2(k,ls));
                    y0=y(k,ls);
                    y00=y1(k,ls);
                end
            end

%************************************************************
%  la méthode de D(k,l)suivant :

            for ls=1:LS
                Sum=0;
                for k=1:m
                    if k~=m;
                        P1(k,ls)=Y1(k+1,ls)-Y1(k,ls);
                        P2(k,ls)=Y2(k+1,ls)-Y2(k,ls);
                        U1=P1(k,ls)/P2(k,ls);
                        U2=P2(k,ls)/P1(k,ls);
                        K12=U1-log(U1)-1;
                        K21=U2-log(U2)-1;
                        Sum=Sum+K12+K21;
                    else
                        P1(k,ls)=1-Y1(k,ls);
                        P2(k,ls)=1-Y2(k,ls);
                        U1=P1(k,ls)/P2(k,ls);
                        U2=P2(k,ls)/P1(k,ls);
                        K12=U1-log(U1)-1;
                        K21=U2-log(U2)-1;
                        Sum=Sum+K12+K21;
                    end
                end
```

```
KK=sqrt(N)*Sum/(m+1);
                KK1(ns,ls)=abs(KK);
            end
        end
%
%*****transformation de notre matrice de KK1 en vecteur:

         % on peut utiliser KKL=reshape(KK1,1,NS*LS)

        KKL=[]
        for i=1:NS

            KKL=[KKL KK1(i,1:LS)];
        end

  KKL=KKL';
%
%*****transformation du vecteur résultat en une matrice:

% on peut utiliser XL=reshape(KKL,L,C)

        ii=1;
        jj=1;
        XL(l,c)=0;
        for nn=1:L
            XL(ii,jj)=KKL(nn);
     % Matrice distorsion lines
            jj=jj+1;
            if jj>c;
                jj=1;
                ii=ii+1;
            end
        end
        XL;
```

```
%*************************************************************
%          Traitement par colonnes
%*************************************************************
*****
C=l*c; % nombre total des éléments de la matrice
n=1;xc(C)=0;
%
%*******transformation de ma matrice en un vecteur ligne:
%
for ii=1:c
    for jj=1:l
        xc(n)=X(jj,ii);                                    %
Vecteur colonnes
        n=n+1;
    end
end
%
%****  Segmentation du vecteur Image et les arranger en matrice
segments suivant les lignes:
%
ls=1;ns=1;xcs(NS,LS)=0;
for l1=1:L1
    xcs(ns,ls)=xc(l1);                                     %
matrice segments
    ls=ls+1;
    if ls>LS
        ls=1;
        ns=ns+1;
    end
end
%
%******Traitement de tous les segments obtenus:
%
for ns=1:NS
    xcss(1:LS)=xcs(ns,1:LS);
%
%****** les vecteurs convolutions obtenus dans les deux cas:
%
    xcss1=conv(xcss,w1);
    xcss2=conv(xcss,w2);
```

```
%
%****calcul de filtre suivant la relation yt=alpha1*yt-1+xt:
%
    for k=1:m;
        y0=1;
        y00=1;
        for ls=1:LS;
            y(k,ls)=conj(Alpha(k))*y0+xcss1(ls+(LS/2-1));
%filtrage pour le premier signal
            A1(k,ls)=mean(y(k,ls)*conj(y0))/mean((abs(y0)).^2);
% auto correlation 1
            y1(k,ls)=conj(Alpha(k))*y00+xcss2(ls+(LS/2-1));
%filtrage pour le 2iéme signal
            Y1(k,ls)=real(exp(-1j*theta)*A1(k,ls));
%autocorrelation modulée

A2(k,ls)=mean(y1(k,ls)*conj(y00))/mean((abs(y00)).^2);%
autocorrélation 2
            Y2(k,ls)=real(exp(-1j*theta)*A2(k,ls));
            y0=y(k,ls);y00=y1(k,ls);
        end
    end

%************************************************************
%  la méthode de D(k,l)suivant :
%
            for ls=1:LS
                Sum=0;
                for k=1:m
                    if k~=m;
                        P1(k,ls)=Y1(k+1,ls)-Y1(k,ls);
                        P2(k,ls)=Y2(k+1,ls)-Y2(k,ls);
                        U1=P1(k,ls)/P2(k,ls);
                        U2=P2(k,ls)/P1(k,ls);
                        K12=U1-log(U1)-1;
                        K21=U2-log(U2)-1;
                        Sum=Sum+K12+K21;
                    else
                        P1(k,ls)=1-Y1(k,ls);
                        P2(k,ls)=1-Y2(k,ls);
                        U1=P1(k,ls)/P2(k,ls);
                        U2=P2(k,ls)/P1(k,ls);
                        K12=U1-log(U1)-1;
                        K21=U2-log(U2)-1;
                        Sum=Sum+K12+K21;
                    end
                end
                KK2=sqrt(N)*Sum/(m+1);
                KK21(ns,ls)=abs(KK2);
            end
end
```

```
%
%*****transformation de notre matrice de KK21 en vecteur:

% on peut utiliser KKC1=reshape(KK21,1,NS*LS)

KKC1=[];
for i=1:NS
    KKC1=[KKC1 KK21(i,1:LS)];
end
KKC1=KKC1';
%
%******transformation du vecteur résultat en une matrice:
% on peut utiliser XC=reshape(KKC1,L,C)

ii=1;jj=1;XC(l,c)=0;
for nn=1:C
    XC(ii,jj)=KKC1(nn);
    ii=ii+1;
    if ii>l;
        ii=1;
        jj=jj+1;
    end
end
XC;
% Matrice distorsion Colonnes
XLC=20*sqrt((abs(XL).^2)+(abs(XC).^2));
XLC1=uint8(XLC);
Tex=toc;

%end
figure(2),clf, subplot(121), imshow(X0),
 title('image originale');
                subplot(122), imshow(XLC1), title('image contour
sans seuillage ');
```

```
% Seuillage de l'image Contour :
%1-Seuillage normal :
    DD1=XLC1;
    MAX_E=max(max(DD1).');
    DD1=DD1/MAX_E;
    threshold=0.5;
 %input('faire entrer le seuil; threshold:= ');

    for m=1:l
        for n=1:c
            if (DD1(m,n)>threshold)
                DD1(m,n)=256;
            else
                DD1(m,n)=0;
            end
        end
    end
    DD=DD1;%DD(:,:,ss)=DD1;
    DDsn=DD;

figure(3),clf, subplot(121), imshow(XLC1),
title('Image Contour avant seuillage');
               subplot(122), imshow(DDsn), title('image contour
après seuillage Normal ');

% 2-Seuillage hystérésis

DD=XLC1;
    DD1=DD;
    MAX_E=max(max(DD1));
    DD1=DD1/MAX_E;
    Sh=0.7;       %input('faire entrer le seuil haut; Sh:= ');
   Sb=0.5;
  %input('faire entrer le seuil bas; Sb:= '); Sb=0.5 ; %

DD2(l,c)=0;DD3(l,c)=0;DD4(l,c)=0;
% seuillage de haut : Sh
    for m=1:l
        for n=1:c
            if (DD1(m,n)>Sh)
                DD2(m,n)=256;
            else
                DD2(m,n)=0;
            end
        end
    end
DDsh=DD2;
```

```
%seuillage en bas Sb :

    for m=1:l
        for n=1:c
            if (DD1(m,n)<Sb )
                DD3(m,n)=256;
            else
                DD3(m,n)=0;
            end
        end
    end
DDsb=DD3;
%seuillage Hybrid (Sh+Sb)
%seuillage en bas Sb :

    for m=1:l
        for n=1:c
            if DD1(m,n)>Sb
            elseif DD1(m,n)< Sh
                        DD4(m,n)=256;
%             else
%                         DD4(m,n)=0;
            end
        end
    end
    DDsbh=DD4;

figure(4),  clf, imshow(DDsh),
title('image à seuillage en haut');
figure(5),  clf, imshow(DDsb),
title('Image à seuillage bas');
figure(6),  clf, imshow(DDsbh),
title('image à seuillage hybride');
figure(7),  clf, imshow(DDsn),
title('Image à seuillage Normal');
```

LISTE DES FIGURES

Figure I.1. Représentation d'une image numérique 5
Figure I.2. Echantillonnage et quantification. 6
Figure I.3. Affichage d'un groupe de pixels 6
Figure I.4. (a) pavage Triangulaire, (b) pavage carré, (c) pavage hexagonal 7
Figure I.5. (a)Maillage hexagonal, (b) Maillage carré, (c) Maillage triangulaire 8
Figure I.6. Connexité des pixels 8
Figure I.7. L'histogramme d'une image noir et blanc. 9
Figure I.8. Histogramme d'image 10
Figure I.9. Utilisation de l'histogramme pour séparer des objets du fond. 10
Figure I.10. Principe d'égalisation d'histogramme 11
Figure I.11.Exemple d'image matricielle 16
Figure I.12.Exemple d'image vectoriel 17
Figure I.13.Exemple d'image en couleur 17
Figure I.14. Exemple d'image binaire 17
Figure I.15. Exemple d'image en niveau de gris 18
Figure I.16. Fenêtre de Hamming (256 samples à 250 kHz, fc=10 kHz) 19
Figure I.17. Fenêtre de Hanning (256 samples à 250 kHz, Fc=10 kHz) 20
Figure I.18. Fenêtre de Blackman (256 samples à 250 kHz, Fc=10 kHz) 20
Figure I.19. Ondelettes Chapeau Mexicain 22
Figure I.20 Ondelettes de Meyer 22
Figure I.21. Ondelettes de Morlet 23

Figure II.1. Quelques modéles countours les plus utilisé est celui en marche d'escalier 25
Figure II.2. Détection de countours (dérivation) :(a): (contour) (b):(Gradient) (c): (Laplacien) 27
Figure II.3. Extraxion des extrema locaux 28

Figure III.1 schéma bloc de filtrage paramétrique. 38
Figure III.2 représentation des deux portes décalées de 50 % de chevauchement. 42
Figure III.3 l'organigramme de segmentation d'images par la méthode (FP). 45
Figure III.4. Image originale(caméra man). 46

Figure III.5. Détection de contour par la méthode PF pour : η_a=-0,3 ,: η_b=0,3 , N=8, m=4, θ= (-pi)..... 47
Figure III.6. Détection de contour par la méthode PF pour : η_a=-0,3 ,: η_b=0,3 , N=16, m=4, θ= (-pi)..... 47
Figure III.7. Détection de contour par la méthode PF pour : η_a=-0,3 ,: η_b=0,3 , N=4, m =2, θ= (- 47
Figure III.8. Détection de contour par la méthode PF pour : η_a=-0.9 ,: η_b=0.9 , N=4, m=2, θ= (-pi)..... 48
Figure III.9. Détection de contour par la méthode PF pour : η_a=-0.9 ,: η_b=0.9 , N=4, m=2, θ= pi..... 48
Figure III.10. Détection de contour par la méthode PF pour : η_a=-0.9 ,: η_b=0.9 , N=4, m=2, θ= 0.025*pi..... 48
Figure III.11. Détection de contour par la méthode DKL et la méthode DLL2 avec seuillage normale..... 49
Figure III.12. Détection de contour par la méthode DKL et la méthode DLL2 avec seuillage hystérésis. 50
Figure III.13. Détection de contour par la méthode DLL2 par l'utilisation de la fenêtre de Hamming ; Hanning ; blackman 51
Figure III.14. Détection de contour par la méthode DLL2 par l'utilisation de l'ondelette Chapeau mexicain ; ondelette Meyer; ondelette Morlet. 51
Figure III.15. Détection de contour par la méthode DKL par l'utilisation de la fenêtre de Hamming, Hanning , blackman 52
Figure III.16. Détection de contour par la méthode DKL par l'utilisation de l'ondelette Chapeau mexicain ; ondelette Meyer; ondelette Morlet. 52

Figure IV.1 schémas explicatifs de la méthode de détection de contours par bloc..... 56
Figure IV.2 lorganigramme de détection de contour d'image par blocs 57
Figure IV.3. Image originale(fruits)..... 58
Figure IV.4. Image originale(lena)..... 60
Figure IV.5. Image originale(terghian) 62

LISTE DES TABLEAUX

Tableau I.1. *Voisinage d'un pixel.* 7
Tableau I.2. *Comparaison entre les formats d'image*.....18

Table des matières

Liste des figures……..I

Liste des tableaux……...II

CHAPITRE I :Généralités sur le traitement d'image…..…….3

I.1. Introduction 4

I.2. Notions générales sur l'image 4

I.2.1. Définition 4

I.2.2. La taille d'une image 5

I .2 .3 Définition d'image numérique 5

I.2.4 Caractéristiques d'une image numérique 6

I.2.5. Pixel: 6

I.2.6 Voisinage, maillage et connexité d'un pixel 7

I.2.7. Résolution 9

I.2.8. Histogramme 9

I.2.9 luminance 11

I.2.10 Contraste 12

I.3. Segmentation 12

I. 3.1. Les différents types de segmentation 12

I.4. Le bruit 14

I.5. Le filtrage 14

I.5.1. Définition 14

I.5.2. Types de filtres 14

I.6 L'autocorrélation 15

I.7. Le seuillage 15

I.8. Types d'images 16

I.8.1. Images matricielles (ou images bitmap) 16

I.8.2. Images vectorielles 17

I.8.3. Images en couleurs 17

I.8.4. Images binaire 17

I.8.5. Images en niveau de gris 18

I .9. Comparaison entre les formats d'image 18

I.10. Fenêtrage 19

I.10.1 Fenêtres courantes 19

I.11. Conclusion 23

CHAPITRE II Techniques de détection de Contours d'image…….... 24

II.1 INRODUCTION …… ……. ……..…… 25

II.2 LE CONTOUR…… …… …… …… …..……..…… …… …… 25

II.3 Principe générale de détection de contour…….26

II.4 Technique linéaire de détection de contour …… ……..…… 26

II.4.1 Approche Gradient…… …… ……...… ...…… 27

II.4.2 Approche Laplace …… …… ……...…..……...…… 28

II.5 Différentes étapes de la détection de contour ……...…… 29

II.6 Les Méthodes classiques de la détection de contour …… …...…… 31

II.6.1 La méthode de CANNY …… 32

II.6.2 La méthode de DERICHE ……..…… 34

II.7 CONCLUSION…… …… …… …… …… 35

CHAPITRE III Détection de contours par la méthode du filtrage paramétrique …….....36

III.1 Introduction 37

III.2 Vue générale sur la méthode du filtrage paramétrique..... 37

III.2.1 Les méthodes à modèle 38

III.2.2 Méthodes sans modèles 38

III.3 La Méthode de Filtrage Paramétrique (PF) 39
III.3.1 Méthode de base 39
III.3.2 Mesures de distorsion 40
III.4 Présentation d'algorithme 41
III.4.1 Bloc de filtrage (le lissage) 41
III.4.2 Bloc d'autocorrélation du premier ordre 43
III.4.3 Bloc de mesure des distorsions 43
III.5 Organigramme de la technique FP 44
III.6 Implémentation de la méthode FP 46
III.6.1 Performance de la méthode FP sans seuillage 46
III.6.2 Performance de la méthode FP avec seuillage 49
III.7 Performance de détection en fonction des diverses fenêtres 50
III.7.1 Pour la méthode DLL2 51
III.7.2 Pour la méthode DKL 52
III.8 Comparaison entre DKL et DLL2 53
III.9 Conclusion 54
CHAPITRE IV Extension de la méthode PF en détection de contour d'image fixe par blocs……..…...55
IV.1 Introduction 56
IV.2 Principe 56
IV.2 organigramme de détection de contour d'image par blocs 57
IV.3 Mise en œuvre et discussion des résultats 58
IV.3.1 premier cas 58
IV.3.2 Deuxième cas 62
IV.4 Conclusion 66
Conclusion Générale……...……67
Bibliographie