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

Université Dr YAHIA FARES de MEDEA



Faculté des Sciences et de la Technologie

Département deGénie Electrique et de l'Informatique

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

Option : Systèmes de Télécommunication

THEME :

Évaluation des méthodes de segmentation en contour d'image fixe

Proposé et dirigé par :
Mr. MIMEN Bakhti

Etudié par :
Mr. .LARBI Farouk
Mr. HAMOUCHE Noureddine

Promotion 2011 /2012

A travers ce modeste travail, nous tenons à remercier vivement notre promoteur « Mr.MIMEN Bakhti » Pour l'intéressante documentation qu'il a mise à notre disposition, Pour ses conseils précieux et pour toutes les commodités et aisances qu'il nous a apportées durant notre étude et réalisation de ce projet.

Nos remerciements les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer notre travail.

Nous exprimons également notre gratitude à tous les enseignants qui ont collaboré à notre formation depuis notre premier cycle d'étude jusqu'à la fin de ce cycle universitaire.

Sans omettre bien sûr de remercier profondément à tous ceux qui ont contribué de près ou de loin à la réalisation du présent travail.



Dédicaces

Dieu me suffit, quel excellent protecteur

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 sœurs, et toute ma famille.

Aussi à ma petite famille de travail.

Et à mes amis et toute personne qui mon aidée de loin ou de prés.

Hamouche noureddine



Dédicaces

Dieu me suffit, quel excellent protecteur

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 sœur,et toute la famille sans exception.

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

mon chère amis bachene youcef

farouk

LISTE DES FIGURES

Numéro de Figure	Titre	Page
Figure I-1	Echantillonnage et quantification.	3
Figure I-2	Résolution spatiale	3
Figure I-3	Résolution tonale	4
Figure I-4	Types de pavage	4
Figure I-5	Types de maillages	4
Figure I-6	Types de connexité	4
Figure I-7	L'histogramme d'une image noir et blanc	5
Figure II.1	Quelques types des contours	9
Figure II.2	Filtrage de contours par seuillage	10
Figure II.3	Rappelé dérive	12
Figure II.4	Image originale	13
Figure II.5	Détecteurs de gradient	13
Figure II.6	Filtre LOG pour deux valeurs	14
Figure II.7	Détecteurs de laplacien différentes de σ	14
Figure III.1	L'organigramme de détection de contour par la méthode FP	21
Figure III.2	Résultat de détection de contours par la méthode de filtrage paramétrique à base des méthodes DKL et DLL2.	24
Figure IV.1	Quelque méthode de segmentation	25
Figure IV.2	Evaluation supervisée	27

LISTE DES TABLEAUX

Numéro dutableau	Titre	Page
Tableau I-1	Quelques types d'images.	2
Tableau I-2	Tableau comparatif	7
Tableau V -1	Critères d'évaluation supervisés	46
Tableau V -2	Résultats d'évaluation de la segmentation de canny	48
Tableau V -3	Résultats d'évaluation de la segmentation par PF D_{KL} et D_{LL}	49
Tableau V -4	Comparaison des résultats de l'évaluation par les différents méthodes	50

ملخص

هذه المذكرة تعتمد بالدرجة الأولى على تحديد الملامح الموجودة في الصورة الثابتة.نركز في عملنا على المقارنة بين طريقة كاني متعددة السلالم و طريقة الترشيح الوسيطي.

طبقنا عدة نظريات لتقييم تجزئة الصور الثابتة المستخلصة من الطريقتين السابقتين .في النهاية نقترح أفضل طريقة للتجزئة

كلمات مفتاح:صورة، ملامح،تجزئة،تقيم ،طريقة كاني ، الترشيح الوسيطي ،منطقة.

.

Résumé :

Ce mémoire est consacré à l'étude et la simulation de la détection des contours des images fixes, basant sur la méthode de canny et la méthode de filtrage paramétrique. Nous avons appliqué plusieurs méthodes d'évaluation de la segmentation d'image fixe sur les deux méthodes précédentes. A la fin nous avons proposé la meilleure méthode de segmentation.

Mots clés : Images ; contours ; segmentation ; évaluation ; méthode de Canny ; filtrage paramétrique ; région.

Abstract :

This thesis is devoted to the study and simulation of edge detection in still images; it is based on the method of canny and parametric filtering method. We applied several types of evaluation criteria for segmentation of still image on the previous two methods. At the end we propose the best method of segmentation.

Keyswords:

Picture; edge; segmentation; evaluation; canny method; Filtering parametric; region.

SOMMAIRE

Dédicace

Remerciement

Résumé iv

Liste des figures iv

Liste des tableaux iv

Sommaire iv

Introduction générale 1

Chapitre I: : Notion de base sur le traitement d'image

I.1 Introduction 2

I.2 Définition de l'image numérique 2

I.3 Caractéristiques d'une image numérique 2

I.3.1 Echantillonnage et quantification 2

I.3.2 Pixel 3

I.3.3 Résolution 3

I.3.4 Pavages du plan 4

I .3.5 Maillages 4

I .3.6 Connexité 4

I .3.7 Histogramme 5

I.4 Segmentation d'image 5

I.5 Egalisation d'histogramme (linéarisation) 5

I.6 Seuillage 6

I.7 Sources de bruit 6

I .7.1 Bruits liés aux conditions de prise de vues 6

I .7.2 Bruits liés au capteur 6

1.7.3 Bruits liés à l'échantillonnage 6

I .7.4 Bruits liés à la nature de la scène 6

I .8 Formats d'images 6

I. 8.1Tableau comparatif 7

I .9 Filtrage numérique 7

I.9.1 Filtre passe-bas (lissage) 7

I.9.2 Filtre passe-haut (Accentuation) 8

I.9.3 Filtrage passe-bande (différentiation) 8

I.9.4 Filtre directionnel 8

I.10. Conclusion 8

Chapitre II : Détection de contours

II.1 Introduction 9

II.2 Contours 9

II.3 Pré- et post traitements 9

II.4 Techniques de détections de contours 10

II.4.1 Filtrage linéaire d'une image 10

II.4.2 Gradient d'image 11

II.4.3 Laplacien d'une image 11

II.5 Approches de détection de contour 12

II.5.1 Approches classiques 12

II.5.1.1 Détecteurs de gradient par filtrage 12

II.5.1.2 Détecteurs de gradient par masques 13

II.5.1.3 Détecteurs de passage par zéro du laplacien 13

II.5.2 Approches analytiques 14

II.5.2.1 Critères de Canny 14

II.5.2.1.1 Bonne détection 15

II.5.2.1.2 Bonne localisation 15

II.5.2.1.3 Non Multiplicité des réponses 15

II.5.2.2 Critères de Deriche 16

II.5.2.3 Critères de Shen et Castan 16

II.7. Conclusion 16

Chapitre III : Détection de contours à base de filtrage paramétrique

III.1 Introduction 17

III.2 Vue générale sur la méthode de Filtrage Paramétrique 17

III .2.1Méthode à modèle 17

III .2.2 Méthodes sans modèle 17

III .3 Méthode de filtrage paramétrique 18

III .3.1 Méthode de base 18

III .3.2 Mesures de distorsion 19

III .4. Application à la détection de contour 20

III .4.1 Points maximaux dans la détection de contour 20

III .4.2 Implémentation logiciel 20

III .4.3 Méthode FP en Bidimensionnel 20

III .5. Résultats pratiques 23

III.6 Conclusion 24

Chapitre IV : Évaluation de la segmentation d'images

IV.1 Introduction 25

IV.2 Différents types d'erreurs de segmentation 26

IV. 2.1 Erreur de sur- segmentation 26

IV. 2.2 Erreur de sous- segmentation 26

IV. 2.3 Erreur de localisation 26

IV. 3Évalaution de la segmentation 26

IV. 3.1 Évaluation visuelle 26

IV. 3.2 Évaluation supervisée 27

IV. 3.2.1 Évaluation avec vérité terrain experte 27

IV. 3.2.2 Évaluation avec vérité terrain synthétique 28

IV. 3.2.2.1 Évaluation de la segmentation en contours 28

IV. 3.2.2.2 Évaluation de la segmentation en régions 30

IV. 3.3 Évaluation non supervisée 31

IV. 3.3.1 Évaluation de la segmentation en contours 31

IV. 3.3.2 Évaluation de la segmentation en régions 31

IV. 4 Conclusion 34

Chapitre V : Implémentation logiciel et évaluation des résultats

V.1 Introduction 35

V.2 Mise en œuvre des algorithmes de détection de contour 35

V .2.1 Algorithme de filtrage paramétrique 35

V .2.1.1 Algorithme de filtrage paramétrique D_{KL} 35

V .2.1.2 Algorithme de filtrage paramétrique D_{LL} 35

V .2.1. 3 Présentation des résultats 36

V .2.2 Algorithme de canny 39

V .2.2.1 Présentation de l'algorithme de canny 39

V .2.2. 2 Présentation des résultats 40

V .3 Évaluation des résultats de segmentation 42

V .3.1 Méthode visuelle 42

V .3.2 Méthode d'évaluation supervisé 45

V .3.2.1Présentation des critères d'évaluation 45

V .3.2.2 Présentation des résultats d'évaluation 46

V .4 Conclusion 51

Conclusion générale 52

Bibliographie 53

Annexes 55

INTRODUCTION GENERALE

L'homme s'est servi de tout temps de représentation graphique pour transmettre des connaissances ou pour noter des situations dont il voulait garder le souvenir, l'utilisation des images se justifié par le fait que le contenu représenté peut être facilement compréhensible par de large classes de personnes possédant se culture d'analyse.

Le traitement d'image a pour vocation l'étude, la conception et la réalisation de système d'exploitation d'images considérées comme véhicule de l'information, son but ultime est d'extraire le contenu informationnel (information pertinente) des images.

De nombreuses recherches on été menées dans le but d'extraire et d'exploiter cette information qui peut être à deux niveaux différents:
-le traitement de bas niveaux.
-le traitement de haut niveau

Dans le traitement de bas niveau, la segmentation d'image est une méthode fondamentale est souvent incontournable fondée sur des critères de similarité et/ou dis similarité d'indices individuels dits<<attributs>>correspondant à des point d'intérêts ou à des zones.

Plusieurs méthodes de segmentation se sont développées ces dernières décennies et les chercheurs ont ressenti le besoin d'avoir une mesure de qualité de la segmentation pour l'évaluation des résultats de la segmentation.

Le cadre général dans le quel s'inscrit ce mémoire est l'évaluation de la segmentation d'image, pour cela nous avons organisé ce mémoire comme suit:

Chapitre1: nous présentons des notions de base sur le traitement d'image.

Chapitre2: nous présentons les différentes techniques de détection de contour.

Chapitre3: nous étudions les différentes critères d'évaluation en contours et en régions.

Chapitre4 : nous présentons la méthode de filtrage paramétrique

Chapitre5: représentation des résultats obtenus par l'utilisation des algorithmes étudiés.

Chapitre I: Notion de base sur le traitement d'image

I.1 Introduction:

Depuis l'image analogique jusqu'à l'image numérique, l'image numérique a changé nos aidées sur le monde d'imagerie. L'informatisation d'image a changé beaucoup de principes et de concepts. Donc avoir des connaissances sur le traitement d'image, permis de nous éclairer et de nous guidé dans ce mémoire, d'où l'intérêt de ce premier chapitre qui contient des brèves définitions, Caractérisation et concepts qui peuvent être utiles.

I .2 Définition d'image numérique :

Une image numérisée est un ensemble de pixels (Picture élément), 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-`a-dire un tableau (matrice) de nombres (binaires). 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 [1].

Phénomène physique	***Grandeur mesurée***	***Capteur***
Émission et réflexion de la lumière visible	Réflectivité, luminance,...	CCD, CMOS, Barrettes CCD,...
Rayonnement Infrarouge	Luminance IR (chaleur), ...	Bolomètres,...
Écho ultra sonore	Distance, densité	Échographie, sonar,...
Résonance magnétique	Présence d'un corps chimique,...	IRM, RMN,...
Écho électromagnétique	Distance, surfaces,..	Radar…
Absorption des rayons X	Densité de tissus,...	Radiographie, tomographie,...

Tab I.1 : Quelques types d'images

CCD : *Charge-Coupled Device*

CMOS : Complementary Metal Oxyde Semi-conductor

IRM : Imagerie par résonance magnétique

RMN : Résonance Magnétique Nucléaire

I.3 Caractéristiques d'une image numérique :

I .3.1 Échantillonnage et quantification :[6]

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

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

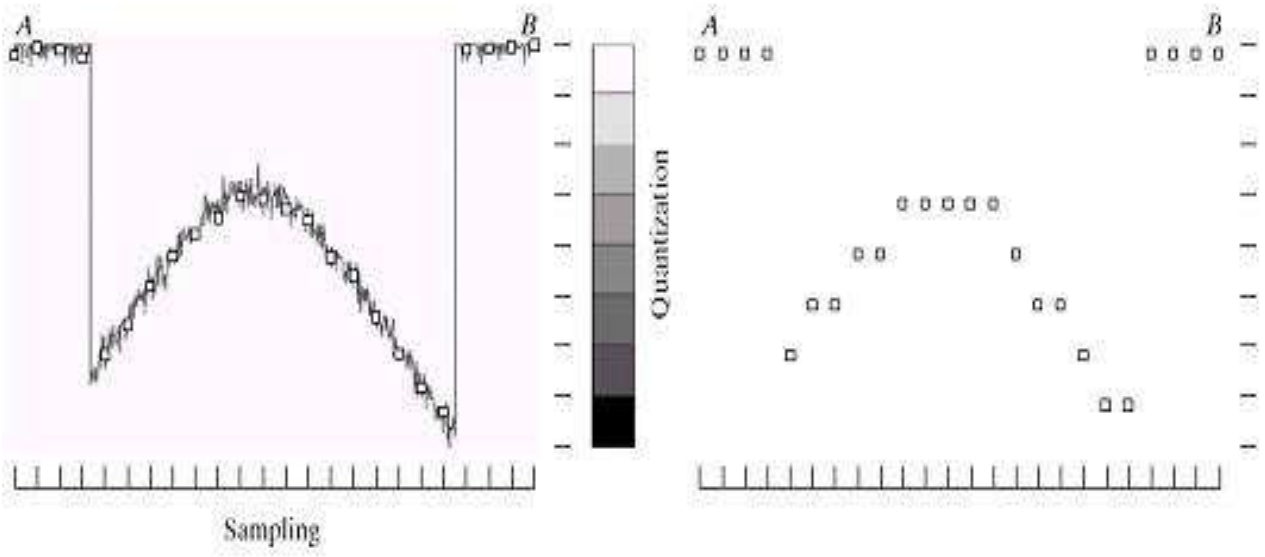


Figure I.1 : échantillonnage et quantification

I .3.2 Définition du PIXEL :[6]

PIXEL est l'abréviation de Picture Elément. C'est le plus petit point de l'écran que l'on peut allumer ou éteindre. Il possède deux caractéristiques :

- Position (coordonnée X,Y)
- nivaux de gris ou Couleur 1, 4, 8, 16 ou 24 bits

I .3.3 La résolution :[6]

La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur. Elle s'exprime le plus souvent en ppp (point par pouces) ou en dpi (dots per inch) parfois en point par cm.

La résolution définit la netteté et la qualité d'une image. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une longueur de 1 pouce), plus votre image est précise dans les détails.

Il ya deux types de résolution :

I .3.3.1 Résolution spatiale : Le plus petit détail discernable



Figure I.2 : Résolution spatiale

I .3.3.2 Résolution tonale :(de tons de gris) : Le plus petit changement discernable

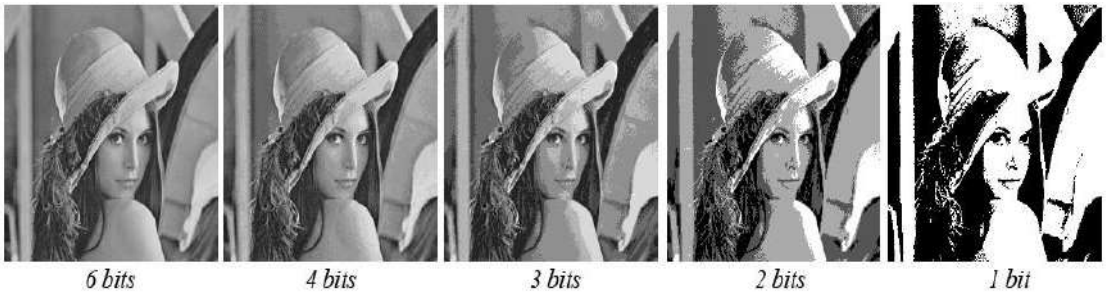
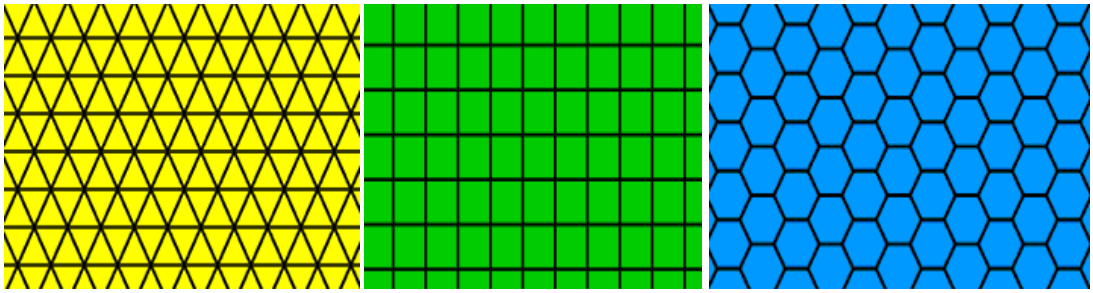


Figure I.3 : Résolution tonale

I .3.4 Pavages du plan :[6]

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

Il n'existe que 3 pavages réguliers du plan :

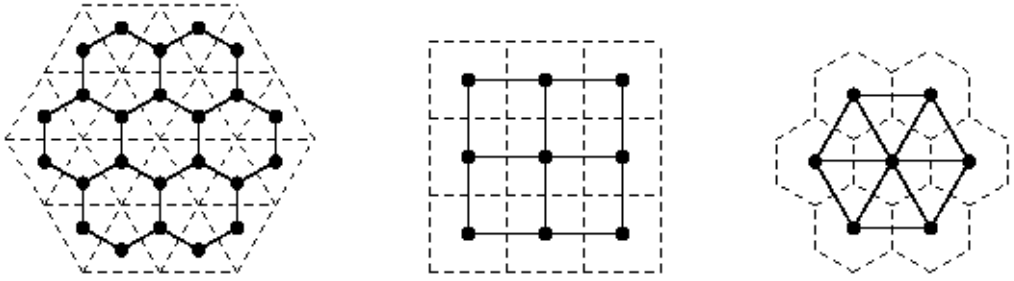


Triangulaire carré hexagonal

Figure I.4 : les types de pavage

I .3.5 Maillages :[6]

A tout pavage du plan on peut associer un graphe où les sommets (nœuds) représentent les cellules élémentaires, et où les arêtes représentent la relation d'adjacence entre les cellules (2 cellules sont adjacentes si elles ont un côté en commun). Un tel graphe est un Maillage du plan.

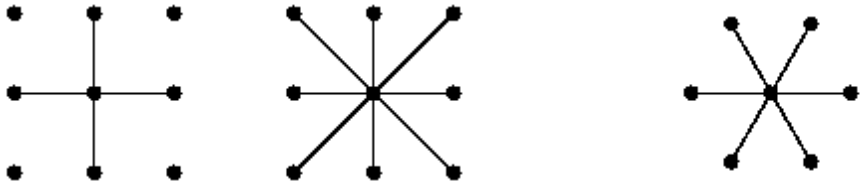


Maillage hexagonal Maillage carré Maillage triangulaire

FigureI.5 : les types de maillages

I .3.6 Connexité[6]

Les relations topologiques dans les images discrètes sont définies à partir de la relation de connexité induite par le graphe du maillage (X,S), où X représente les sommets et S les arêtes.



maille carré 4-connexe | maille carrée 8-connexe | maille triang. 6-connexe

Figure I.6 : les types de connexité

I .3.7 L'histogramme

L'histogramme est la représentation graphique de la proportion de pixels d'une image par gamme de luminosité. Au delà de cette définition un peu barbare, 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.

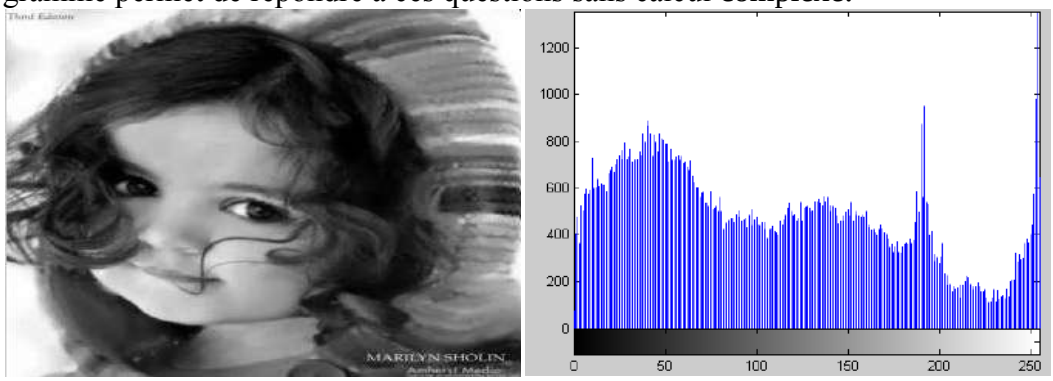


Figure I.7 : L'histogramme d'une image noir et blanc

I .4 La segmentation d'une image :

C'est une partition de l'ensemble de pixels de l'image en différents groupes, chaque groupe est supposé correspondre à un "objet" de l'image.

Quand il s'agit de séparer les pixels en 2 groupes, on parle alors de binarisation. il n'existe pas une méthode de segmentation d'images, mais plusieurs ; où la méthode utilisée dépend très fortement du type d'images et de l'application visée.

I .5 Égalisation d'histogramme (linéarisation) [6]:

Le but d'égalisation d'histogramme est de rendre l'histogramme le plus plat possible :

- Distribution uniforme des niveaux de gris.
- Maximisation de l'entropie de l'image.
- Image donnant une information maximale.

I .6 Opération du seuillage[6] :

A partir des histogrammes unidimensionnels de l'image, on peut aisément pratiquer une segmentation, voire une classification rudimentaire et cela par une opération toute simple appelée le seuillage.

Le seuillage effectué avec un seuil unique revient à une opération de binarisation. Soit I(x,y), la valeur radiométrique du pixel de coordonnées (x,y) dans l'image, si I(x,y) > T, T étant le seuil fixé, le pixel prend la valeur 1, sinon il prend la valeur 0. Les pixels ont été classés en deux catégories, on a créé une image binaire en noir et blanc.

I .7 Les sources de bruit :[13]

I .7.1 Bruits liés aux conditions de prise de vues :

Ce sont des événements vérifiant les conditions d'acquisition du signal :

- Le bougé
- Problèmes liés à l'éclairage de la scène observée

I .7.2 Bruits liés au capteur :

- Capteur mal réglé
- Capteur de mauvaise qualité (distorsion de la gamme des niveaux de gris ou en flou)

I .7.3 Bruits liés à l'échantillonnage :

Ces bruits reflètent essentiellement des problèmes de quantification

- précision d'environ
- problèmes dans le cas d'applications de grande précision

I .7.4 Bruits liés à la nature de la scène :

Exemples :

- Nuage sur les images satellitaires
- Poussières dans les scènes industrielles
- Brouillard pour les scènes routières

I .8 Formats d'images :[extrait de l'encyclopédie wkipedia]

Un format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

La plupart des formats sont composés d'un en-tête contenant des attributs (dimensions de l'image, type de codage, etc.), suivi des données (l'image proprement dite). La structuration des attributs et des données diffère pour chaque format d'image. De plus, les formats actuels intègrent souvent une zone de métadonnées (metadata en anglais) servant à préciser les informations concernant l'image comme : la date, l'heure et le lieu de la prise de

vue, les caractéristiques physiques de la photographie (sensibilité ISO, vitesse d'obturation, usage du flash.

I .8.1Tableau comparatif

	Type (matriciel/ vectoriel)	Compression des données	Nombre de couleurs supportées	Affichage progressif	Animation	Transparence
JPEG	matriciel	Oui, réglable (avec perte)	16 millions	Oui	Non	Non
JPEG2000	matriciel	Oui, avec ou sans perte	4 milliards	Oui	Oui	Oui
GIF	matriciel	Oui, Sans perte	256 maxi (palette)	Oui	Oui	Oui
PNG	matriciel	Oui, Sans perte	Palettisé (256 couleurs ou moins) ou 16 millions	Oui	Non	Oui (couche Alpha)
TIFF	matriciel	Compression ou pas avec ou sans pertes	de monochrom e à 16 millions	Non	Non	Oui (couche Alpha)
SVG	vectoriel	compression possible	16 millions	ne s'applique pas	Oui	Oui (par nature)

Tab I.2 : Tableau comparatif

I .9 Filtrage numérique :

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage**.**

Le filtrage consiste à modifier la distribution fréquentielle des composantes d'un signal selon des spécifications données.

Le système linéaire utilisé est appelé filtre numérique.

Parmi ces systèmes, nous distinguons : les filtres passe-bas (lissage), filtres passe-haut (accentuation), filtres passe-bande (différenciation) et filtres directionnels [5] [12].

I.9.1 Filtre passe-bas (lissage) :

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

L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image. Elle peut être répétée plusieurs fois, ce qui crée un effet de flou. En pratique, il faut choisir un compromis entre l'atténuation du bruit et la conservation des détails et contours significatifs [5][12].

I.9.2 Filtre passe-haut (Accentuation) :

Le renfoncement des contours et leur extraction s'obtiennent dans le domaine fréquentiel par l'application d'un filtre passe-haut.

Le filtre digital passe-haut a les caractéristiques inverses du filtre passe-bas .Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence [5][12].

I.9.3 Filtrage passe-bande (différentiation) :

Cette opération est une dérivée du filtre passe-bas. Elle consiste à éliminer la redondance d'information entre l'image originale et l'image obtenue par filtrage passe-bas. Seule la différence entre l'image source et l'image traitée est conservée [5][12].

I.9.4 Filtre directionnel :

Dans certains cas, on cherche à faire apparaître des détails de l'image dans une direction bien déterminée. Pour cela, on utilise des filtres qui opèrent suivant des directions (horizontales, verticales et diagonales)[5][12]**.**

I.1. Conclusion :

On trouve ses applications dans des domaines très variés tels que la télécommunication (T.V., vidéo, publicité,...), médecine (radiographie, ultrasons,...), biologie, l'astronomie, géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, et l'armement (application militaire)**.**

Dans ce mémoire le traitement d'image nos serai plus utile pour l'évaluation de la segmentation d'image.

Chapitre II : détection de contours

II.1 Introduction :

La détection de contours dans les images a débuté de façon extrêmement empirique par des opérateurs locaux qui, soit estimaient un gradient, soit convolaient l'image par des masques caractéristiques des contours. Dans les années 80, des approches plus systématiques ont été mises en place par Marr , puis Canny , pour obtenir des contours plus significatifs. Ces travaux ont abouti à une bonne compréhension de ce qu'il faut faire pour détecter les contours, mais la définition même des contours demeure très vague, ce qui rend ces techniques encore peu efficaces sur un problème concret. De plus, on a pu montrer que le problème de détection de contours est généralement mal posé (au sens de la résolution des systèmes) . Les seuls modèles de contours utilisables sont ceux de contours idéalisés.

C'est pourquoi, même si de très gros progrès ont été accomplis dans ce domaine, les techniques empiriques d'estimation du gradient proposées dans les années 70-80 restent souvent encore employées en concurrence de techniques plus modernes. Une excellente référence à ce problème est l'ouvrage collectif .Plusieurs articles existent sur la comparaison des performances de détecteurs de contours, en particulier [4].

II.2 Les contours :[7] [8]

Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de réflectance (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

Et sont caractérisés par des discontinuités de la fonction d'intensité dans les images. Le principe de la détection de contours repose donc sur l'étude des dérivées de la fonction d'intensité dans l'image : les extrema locaux du gradient de la fonction d'intensité et les passages par zéro du laplacien.

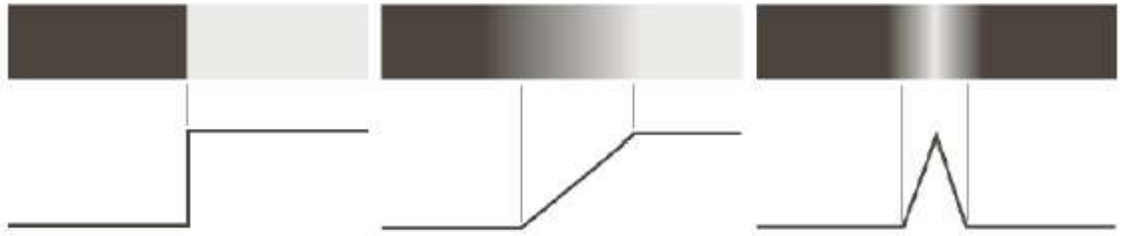


Figure II.1: Quelques types des contours

II. 3 Pré- et post traitements :[8]

La qualité de la détection est très liée à la qualité des contours dans l'image (cela sera vérifié pour toutes les méthodes). Ainsi, elles sont souvent bonnes pour les applications de robotique, en environnement artificiel, pour des images aériennes à forte résolution (en milieu urbain par exemple), elles se dégradent très vite pour les scènes d'extérieur, les images

médicales, et deviennent totalement inapplicables en imagerie bruitée (thermographie, imagerie ultrasonore, radar, etc.).

Ces détecteurs doivent toujours être suivis d'une étape de post-filtrage, et souvent précédés d'une étape de pré-filtrage.

Parmi les filtres de pré-traitement, on recherche ceux qui diminuent le bruit tout en préservant les discontinuités. Filtre médian par exemple, filtre toboggan, ou filtre de diffusion anisotrope de Perona et Malik . Ces filtres ont cependant le défaut de créer des contours artificiels qui peuvent être par la suite difficiles à éliminer.

Les post-traitements commencent le plus souvent par l'élimination des points de contour trop faibles par un seuillage (seuillage adaptatif si l'image est très hétérogène). On élimine ensuite les points qui ne sont pas des extrémums locaux dans la direction du gradient.

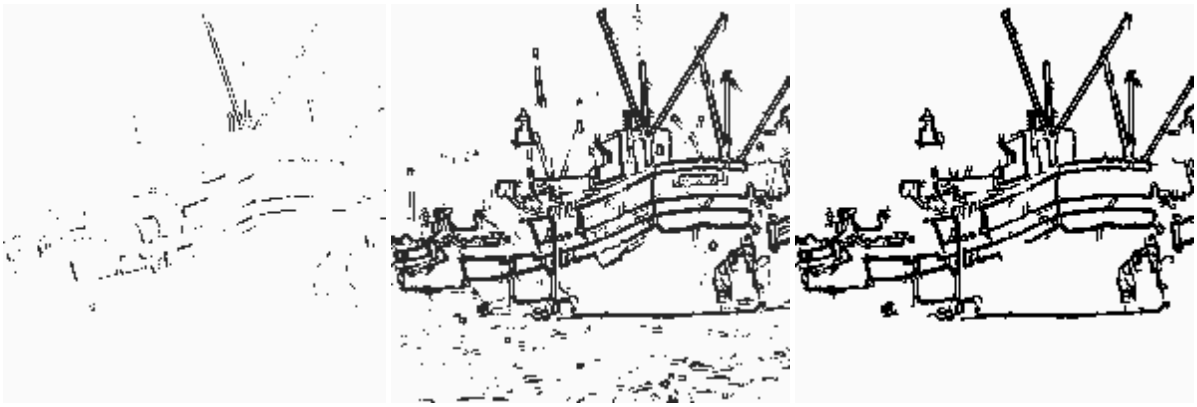


Figure II.2: Filtrage de contours par seuillage

Enfin on procède à des étapes de poursuite et de fermeture des contours. La première opération a pour objectif de supprimer les petites disparitions de contours qui peuvent se produire par suite du bruit ou d'une perte de contraste, la seconde se propose de détecter des objets topo logiquement clos, c'est-à-dire des zones fermées.

II.4 Les techniques de détections de contours :[8] [9]

II.4.1 Le filtrage linéaire d'une image :

Filtrer une image consiste à convoler sa fonction d'intensité I(x, y) avec une fonction h(x, y) appelée réponse i pulsionnelle du filtre.

$$I'(x,y) = h(x,y) * I(x,y) \qquad \text{II.1}$$

$$I'(x,y) = \iint_{-\infty}^{+\infty} h(u,v) I(x-u, y-v)\,du\,dv \qquad \text{II.2}$$

$$I'(x,y) = \iint_{-\infty}^{+\infty} h(x-u, x-v) I(u,v)\,du\,dv \qquad \text{II.3}$$

Dans le cas discret :

$$I'(x,y) = \sum_{u=-H/2}^{+H/2} \sum_{v=-H/2}^{+H/2} h(u,v) I(x-u, y-v). \quad \text{II.4}$$

où H correspond à la dimension du masque de filtrage.

II.4.2 Le gradient d'une image :[7]

Le gradient d'une image est le vecteur ∇I(x, y) défini par :

$$\nabla I(x,y) = \left(\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right) \quad \text{II.5}$$

Il est donc caractérisé par un module *m* et une direction ∅ dans l'image :

$$m = \sqrt{\frac{\partial I(x,y)^2}{\partial x} + \frac{\partial I(x,y)^2}{\partial y}}, \quad \text{II.6}$$

$$\emptyset = arctan\left(\frac{\partial I(x,y)}{\partial x} / \frac{\partial I(x,y)}{\partial y}\right) \quad \text{II.7}$$

La direction du gradient maximise la dérivée directionnelle.

La dérivée de *I(x, y)* dans une direction donnée ***d*** s'écrit :

$$\nabla I(x, y) \, .d.$$

Le gradient d'une image filtrée :

$$\nabla I'(x, y) = \nabla (I(x, y) * h(x, y)) = \nabla I(x, y) * h(x, y) = I(x, y) * \nabla h(x, y). \quad \text{II.8}$$

II.4.3 Le Laplacien d'une image :[7]

Le laplacien d'une image d'intensité *I(x, y)* est défini par :

$$\nabla^2 I(x,y) = \left(\frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2}\right) \quad \text{II.9}$$

Invariant aux rotations de l'image.

Le Laplacien est souvent utilisé en amélioration d'images pour accentuer l'effet de contour :

$$I'(x, y) = I(x, y) - c\nabla^2(x, y). \quad \text{II.10}$$

Sensibilité au bruit accrue par rapport au gradient.

Le Laplacien d'une image filtrée :

$$\Delta I'(x, y) = \Delta I(x, y) * h(x, y) = I(x, y) * \Delta h(x, y). \quad \text{II.11}$$

fonction

Le gradient

Le laplacien

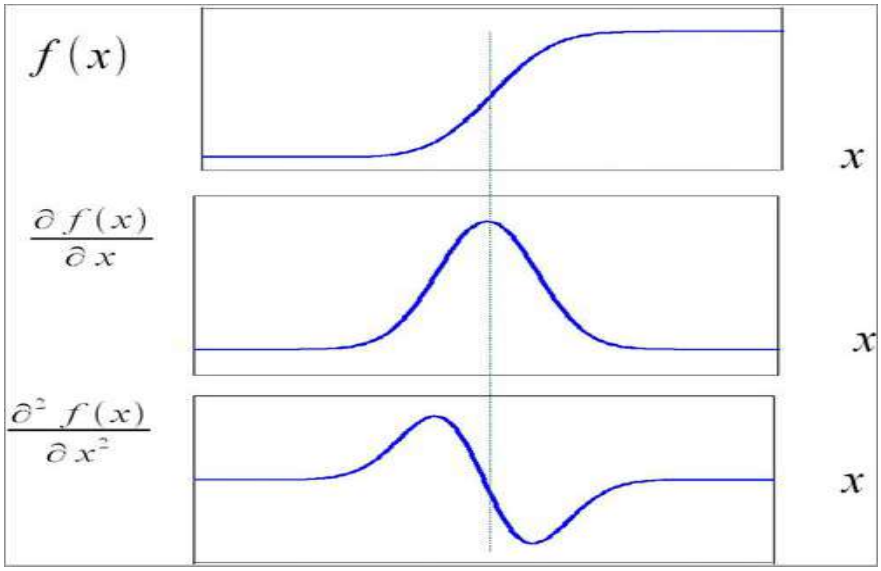


Figure II.3: Rappelé dérive

II.5 Les approche de détection de contour:[8]

II.5.1 Les approches classiques :

Un ensemble de méthodes qui ont eu historiquement une grande importance en traitement des images. Bien que certaines soient encore régulièrement employées.

II.5.1.1 Les détecteurs de gradient par filtrage :

Ces détecteurs reposent tous sur une recherche d'un extremum de la dérivée première, celle-ci étant calculée de diverses manières, mais généralement par un filtrage passe haut précédé d'un léger filtrage passe-bas pour s'affranchir des bruits. Ces approches par filtrage linéaire sur des critères très simples de sélection en fréquence, par exemple un filtre de base de Modestino et Fries est un filtre récursif de la forme :

$$H_0(z_1, z_2) = \frac{\sum_i^n \sum_j^n b_{i,j} z_1^{-i} z_2^{-j}}{1+\sum_i^n \sum_j^n a_{i,j} z_1^{-i} z_2^{-j}} \qquad \text{II.12}$$

Les coefficients $a_{i,j}$ et $b_{i,j}$ sont déterminés à partir de 3 paramètres spécifiques de l'image à traiter :

γ : le paramètre poissonnier de la densité de contours.

ρ : la constante de la loi exponentielle de l'auto-corrélation de l'image.

ϵ : la puissance du bruit.

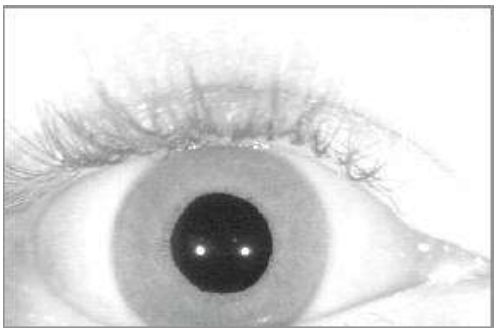


Figure II.4: Image originale

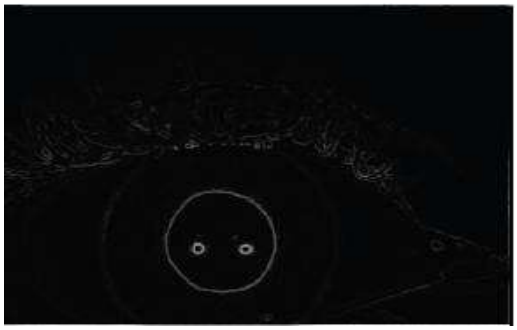
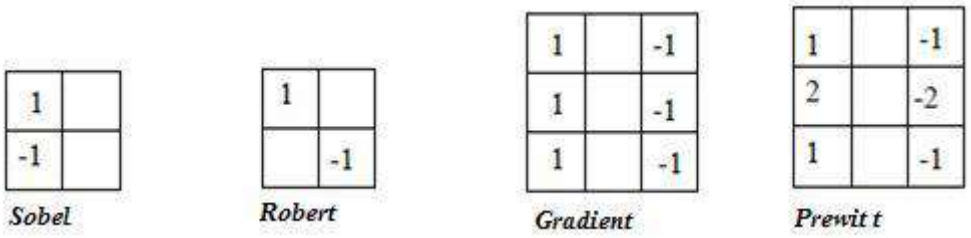


Figure II.5 : Détecteurs de gradient

II.5.2 Les détecteurs de gradient par masques :[7]

Les filtres de dérivation plus empiriques ont été proposés à partir d'estimateurs locaux de l'image f ou de ses dérivées $\frac{\partial f}{\partial x}$. Ces estimées sont obtenues à l'aide de masques appliqués sur des fenêtres de 2*2pixels ou 3*3 (exceptionnellement, en cas d'images très bruitées, sur des fenêtres plus grandes). On note sans surprise que la somme des coefficients de ces filtres est nulle (fonction de transfert nulle à la fréquence 0), et que les coefficients sont antisymétriques. Les filtres les plus utilisés sont, dans l'ordre décroissant :

Sobel- Roberts-Gradient- Prewitt.



Les filtres 3*3 sont un peu moins précis (c'est-à-dire que les contours qu'ils détectent sont moins bien localisés et souvent épais), mais les images ainsi obtenues sont généralement plus fiables et permettent des post-traitements plus poussés, ils sont également centrés sur un pixel et non entre des pixels (filtres à phase nulle).

II.5.3 Les détecteurs de passage par zéro du Laplacien :

Elles utilisent le fait que le passage par zéro du laplacien permet de bien mettre en évidence les extrémums de la dérivée, le passage par zéro du laplacien correspond en effet bien au maximum du gradient dans la direction du gradient. Ces méthodes tirent en outre profit du fait que les zéros de la dérivée seconde constituent un réseau de lignes fermées (évitant donc, en principe, les étapes de poursuite et de fermeture). Il en est de même du réseau des lignes de crête du gradient, mais le premier est plus aisément détecté à partir d'un simple étiquetage des zones positives et négatives. Mais l'estimation de la dérivée seconde étant très sensible aux bruits, il convient de filtrer très fortement l'image avant de mesurer le laplacien. Cela conduit au filtrage suivant :

$$\psi * \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)[\psi * f] \qquad \text{II.13}$$

Ou ψ est un filtre passe-bas ; ce qui se récrit symboliquement :

$$\text{Image des contours} = \text{passage par zéro } \left(f * \left(\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2}\right)\right)$$

Cela exprime qu'une image de contours est obtenue par filtrage de l'image par la dérivée seconde d'un filtre passe bas, puis détection des zéros de la fonction ainsi obtenue.

Les filtres les plus utilisés pour ces filtrages passe-bas sont les filtres gaussiens . Marr a montré en effet qu'avec de tels filtres on pouvait approcher de très prés les effets donnés par le système visuel humain. Le filtre obtenu par convolution avec le laplacien d'une gaussienne est connu sous le nom de LOG. [7]

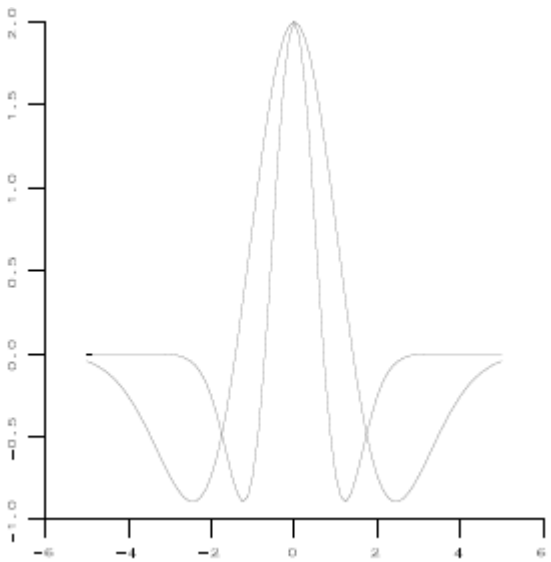


Figure II.6: Filtre LOG pour deux valeurs

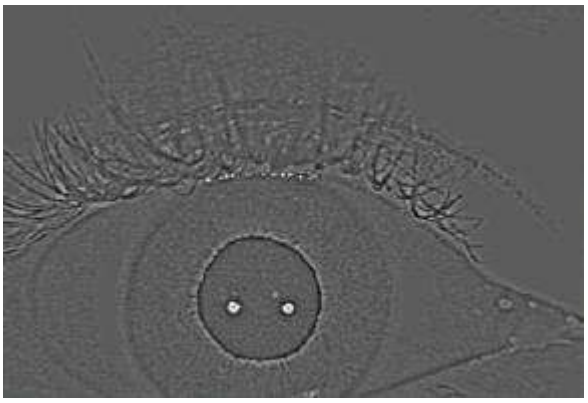


Figure II.7: Détecteurs de laplacien différentes de σ

II.6 Les approches analytiques :

Nous allons voir maintenant une approche qui a permis une bien meilleure compréhension des conditions d'une bonne détection de contours et qui a ainsi conduit à des détecteurs de très bonne qualité.

II.6.1 Les critères de Canny :[8]

Parmi les approches par optimisation, celle développée par Canny semble être la plus intéressante et mérite incontestablement une présentation plus en détail. Soit *I(x)* un signal monodimensionnel représentant un saut d'amplitude A, noyé dans un bruit Gaussien *n(x)* de moyenne nulle et de variance n_0^2 :

$$I(x) = A\, u_{-1}(x) + n(x) \qquad \text{II.14}$$

Ou u_{-1} désigne la fonction de Heaviside.

Soit$\Theta(x0)$, la sortie au point x0 de la convolution du signal *I(x)* avec un opérateur de détection *f(x)* :

$$\Theta(x_0) = \int_{-\infty}^{+\infty} I(x) f(x_0 - x) dx \quad \text{II.15}$$

Le problème posé est de trouver *f(x)* tel que $\Theta(x_0)$soit maximum sous les 3 contraintes suivantes :

II.6.1.1. Bonne détection :

Ce critère revient à chercher f(x) antisymétrique et tel que le rapport signal sur bruit ∑ soit maximum:

$$\sum = \frac{A \int_{-\infty}^{0} f(x) dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad \text{II.16}$$

II.6.1.2 Bonne localisation :

Ce critère corresponds à la minimisation de la variance σ^2de la position des passages par zéro et revient à maximiser la localisation Λ définie comme l'inverse de σ^2 :

$$\Lambda = \frac{A|f'(0)|}{n_0 \sqrt{\int_{-\infty}^{+\infty} f'^2(x) dx}} \quad \text{II.17}$$

II.6.1.3 Non Multiplicité des réponses:

Ce critère corresponds à la limitation du nombre de maximas locaux détectés en réponse à un seul contour. La distance moyenne entre les maximas locaux , notée x_{max} est alors contrainte à l'égalité suivante :

$$x_{max} = 2\pi \left(\sqrt{\frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''^2(x) dx}} \right) \quad \text{II.18}$$

Trouver *f(x)* qui maximise le produit ∑Λ sous la contrainte que le 3ième critère soit fixé à une constante k revient alors à trouver la solution de l''equation différentielle suivante :

$$2f(x) - 2\lambda f'(x) + 2\lambda_2 f'' n(x) + \lambda_3 = 0 \quad \text{II.19}$$

Qui admet comme solution générale :

$$f(x) = a_1 e^{\alpha.x} \sin(w.x) + a_2 e^{\alpha.x} \cos(w.x) + a_3 e^{-\alpha.x} \sin(w.x) + a_4 e^{-\alpha.x} \cos(w.x) \quad \text{II.20}$$

Cherchant l'opérateur *f(x)* sous forme de filtre à réponse impulsionnelle finie (RIF) définie sur l'intervalle [−W,+W] et présentant une pente S à l'origine, Canny a imposé les conditions aux limites suivantes :

$$f(0) = 0 \qquad f(W) = 0 \qquad f'(0) = S \qquad f'(W) = 0$$

II.6.2 Les critères Deriche :[8]

L'approche développée par Deriche a consisté en la recherche de l'opérateur optimal sous forme de filtre à réponse impulsionnelle infinie (RII). Il en résulte une équation différentielle admettant comme solution l'opérateur suivant :

$$f(x) = \frac{s}{w} e^{-\alpha|x|} \sin(wx) \qquad \text{II.21}$$

Évaluant les différentes intégrales intervenant dans le calcul des indices de performance de cet opérateur, on obtient les résultats suivants :

$$\Lambda = \sqrt{2\alpha} \quad \sum = \sqrt{\frac{2\alpha}{\alpha^2 + w^2}} \qquad \text{II.22}$$

$$\sum \Lambda = \frac{2\alpha}{\sqrt{\alpha^2 + w^2}} \quad k = \sqrt{\frac{\alpha^2 + w^2}{5\alpha^2 + w^2}} \qquad \text{II.23}$$

II.6.3 Les critères Shen et Castan :[9]

L'approche développée par Shen et Castan a consisté en l'élaboration d'un filtre passe bas optimal pour la détection des contours. Le formalisme mathématique développé s'inspire fortement de celui de Canny. Un modèle de saut d'amplitude bruité est considéré et un critère à optimiser est alors dérivé. Il en ressort de cet étude que le filtre linéaire optimal de lissage recherché est le filtre exponentiel symétrique :

$$s(x) = ae^{-\alpha|x|} \qquad \text{II.24}$$

Qu'ils proposent de mettre en œuvre comme suit :

$$y_1(n) = x(n-1) + e^{-\alpha} y_1(n-1) \quad \text{pour n=1,…, M} \qquad \text{II.25}$$

$$y(n) = y_1(n+1) + e^{-\alpha} y(n+1) \quad \text{pour n=1,…, M} \qquad \text{II.26}$$

II.7 Conclusion :

Afin de détecter des contours, nous venons d'utiliser des opérations de bas niveau telles que le filtrage linéaire mais les résultats sont imparfaits pour diverses raisons. Ce qui rend cette détection difficile à ce niveau provient de la présence du bruit, de la difficulté de régler les paramètres du détecteur de manière optimale, la présence de contraste variable.

Chapitre III : Détection de contours à base de filtrage paramétrique

III .1Introduction :

À l'aide de l'analyse de l'autoccorrélation du premier ordre du signal image filtré, les chercheurs produiront une 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. Cette nouvelles techniques de détection de contours est appelée **'filtrage paramétrique'** cette méthode est appliquée avec succès sur la parole et la segmentation, Cette nouvelle approche ouvre des voies très prometteuses dans l'analyse de contour et la segmentation.

III .2 Vue générale sur la méthode de Filtrage Paramétrique

III .2.1Méthode à modèle :[10]

Elles dépendent essentiellement des modélisations LPC(Linear Predictive Coding). Une des méthodes est dérivée d'un test GLR (a 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. 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.

III .2.2 Les méthodes sans modèles :[10]

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** : $D = \sup\left|F_1(\omega) - F_2(\omega)\right|$ et la distance spectrale de **Cramer-Von Mises** : $D = \int \left|F_1(\omega) - F_2(\omega)\right|^2 d\omega$ ou $F_1(\omega)$ et $F_2(\omega)$ sont des fonctions de distributions spectrales estimées des deux domaines du signal pris dans le voisinage du temps 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)

$$D_{KL} = \int_{-\pi}^{\pi} K\left[\frac{f_1(\omega)}{f_2(w)}\right] d\omega \qquad \text{III.1}$$

Avec

$$K(u) = u - \log(u) - 1 \qquad \text{III.2}$$

Et la distance L2 de log-Spectrum :

$$D_{LL2} = \int \left|\log f_1(\omega) - \log f_2(w)\right|^2 dw \qquad \text{III.3}$$

Ou $f_1(\omega)$ 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 :[10]

La méthode de filtrage paramétrique (FP) 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.

III .3.1La 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 :

$$\rho_k = E\left(X_{t+k} X_t\right) \Big/ E(X_t^2) \qquad \text{III.4}$$

Considérons le filtre récursive IIR $H(z^{-1};\alpha)$ définie par :

$$Y_t(\alpha) = \sum_{l=0}^{\infty} \overline{\alpha}^l X_{t-1} = \overline{\alpha} Y_{t-1}(\alpha) + X_t \qquad \text{III.5}$$

Ou : $\alpha = \eta e^{-j\theta}$ est un nombre complexe avec : $|\eta| < 1 \text{ et } \theta \in [-\pi, \pi]$ et $\overline{\alpha}$ représente le conjugué de α .Soit $\rho(\alpha)$ l'autocorrélation d'ordre un de $\{Y_t(\alpha)\}$

$$\rho(\alpha) = \frac{E\left(Y_{t+1}(\alpha) \cdot \overline{Y}_t(\alpha)\right)}{E\left(\left|Y_t(\alpha)\right|^2\right)} \qquad \text{III.6}$$

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

$$\gamma_\theta(\eta) = \Re\left(e^{-j\theta} \rho(\alpha)\right) \qquad \text{III.7}$$

Ou : $\Re(.)$ représente la partie réelle d'un nombre Complexe. 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 la méthode de 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, ou les η_k peuvent être pris par exemple, uniformément dans l'intervalle $[\eta_a, \eta_b] \subset (-1,1)$. Pour chaque évaluation de $\gamma_\theta(\eta)$, le nombre de multiplications exigé est proportionnelle 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 :[11]

Étant donné deux domaines (ou plages) du signal image à traiter, nommés $X_t^{(1)}$ et $X_t^{(2)}$, 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. 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 significatif de changements da la structure de corrélation.

La méthode de 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 a dire:

$$\gamma_\Omega^p = \left(\int_\Omega \left| \gamma_\theta^{(1)}(\eta) - \gamma_\theta^{(2)}(\eta) \right|^p d\theta \, d\eta \right)^{1/p} \qquad \text{III.8}$$

Pour $p \in (0, \infty)$, et la distance KL définie par

$$\hat{K}_\Omega = \int_\Omega \left(K\left[\frac{p_\theta^{(1)}(\eta)}{p_\theta^{(2)}(\eta)} \right] + K\left[\frac{p_\theta^{(2)}(\eta)}{p_\theta^{(1)}(\eta)} \right] \right) d\theta \, d\eta \qquad \text{III.9}$$

Ou

$$\hat{K}_\Omega = \int_\Omega \left(p_\theta^{(2)}(\eta) K\left[\frac{p_\theta^{(1)}(\eta)}{p_\theta^{(2)}(\eta)} \right] + p_\theta^{(1)}(\eta) K\left[\frac{p_\theta^{(2)}(\eta)}{p_\theta^{(1)}(\eta)} \right] \right) d\theta \, d\eta \qquad \text{III. 10}$$

Dans cette expression Ω est un sous ensemble $[-\pi, \pi] \times [\eta_a, \eta_b]$, $\gamma_\theta^{(1)}(.)$ et $\gamma_\theta^{(2)}(.)$ sont les fonction caractéristique obtenue de $X_t^{(1)}$ et $X_t^{(2)}$, $p_\theta^{(1)}(.)$ and $p_\theta^{(2)}(.)$ sont les fonctions densités définie dans l'intervalle $[\eta_a, \eta_b] \subset (-1,1)$, ayant la forme :

$$p_\theta(\eta) = \frac{1}{2}\Big(\frac{d\gamma_\theta(\eta)}{d\eta} + [\gamma_\theta(\eta_a) + 1]\delta(\eta - \eta_a) + [1 - \gamma_\theta(\eta_b)]\delta(\eta - \eta_b) \Big) \qquad \text{III.11}$$

Dans l'application ces mesures de distorsion sont discrétiser en utilisant la sortie $\gamma_\theta(\eta)$ du filtre de la figure suivant.

III .4. Application à la détection de contour :[11]

III .4.1 Les points maximaux dans la détection de contour :

Dans notre expérience préliminaire de la détection de contour de l'image, l'approche de choix du pic est employée pour la détection.Une méthode typique de choix du pic prend des portes (ou des fenêtres) de **2N Points** du signal image original, en chaque point 't' ; une fenêtre (porte) est centrée en '**t**' et l'autre et décalée avant de **m-points.**

Pour la suite de toutes les expériences, on prend **m=N/2 (**i.e. 50% de *chevauchement*). Les deux portes sont multipliées par *la fenêtre de Hamming* de Points (*Hamming Window*) avant de les utiliser dans l'évaluation des mesures de distorsions. Soit D_t le résultat de distorsions ; les localisations de changement spectrales. Ces localisations peuvent être identifiées par le passage par zéro (*the zero croissing*) de la différence $\mathbf{D_t}$**-**$\mathbf{D_{t-1}}$**.** Un pic de $\mathbf{D_t}$ est considéré comme significatif si son amplitude dépasse un seuil **T.**

Une performance améliorée peut être obtenue en général par le lissage (*smoothing*) de la trajectoire de $\mathbf{D_t}$ avant le pas de choix du pic.

III .4.2 L'implémentation de logiciel :[10]

La démarche suivie pour l'implémentation de logicielde la méthode PF pour la détection de contour est illustrée dans la figure suivant :

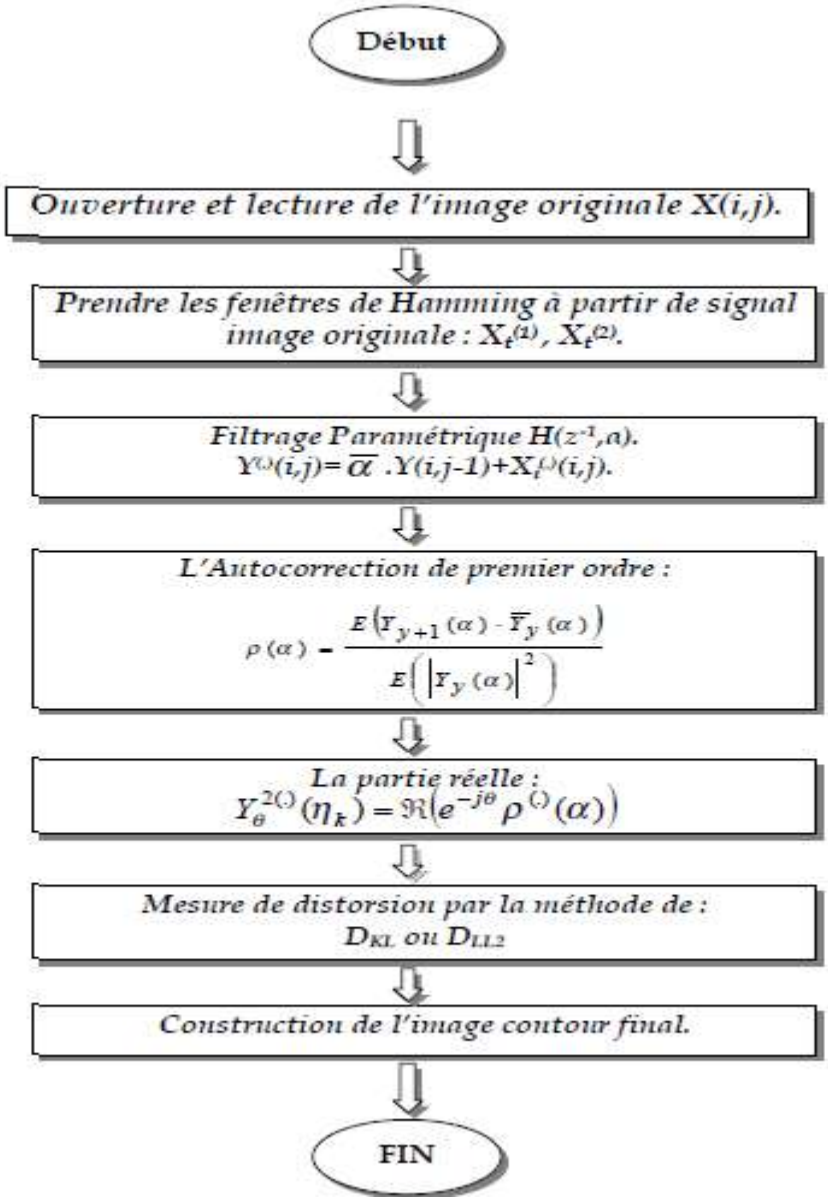


Figure III.1 : L'organigramme de détection de contour par la méthode FP

III .4.3 La Méthode FP en Bidimensionnel :[10]

On applique cette technique (méthode PF) sur une image I(x,y), donc, on va suivre ces étapes :

1. On va calculer le paramètreα: $\alpha_k=\eta_k e^{j\theta}$
   Avec

$$\eta_k=\eta_a+\frac{(k-1)(\eta_b-\eta_a)}{(m-1)} \quad \text{pour } k=1,\ldots,m \qquad \text{III.12}$$

2. On filtrer l'image*I(x, y)* suivant les lignes *(X)*:

$$Y_x^{(.)} = \overline{\alpha}.Y_{x-1}^{(.)} + X_x^{(.)} \qquad \text{III.13}$$

3. L'autocorrélation suivant les lignes *(X)*:

$$\rho(\alpha) = \frac{E\left(Y_{x+1}(\alpha)\cdot \overline{Y}_x(\alpha)\right)}{E\left(\left|Y_x(\alpha)\right|^2\right)} \qquad \text{III.14}$$

Et ensuite, on prend:

$$Y_\theta^{2(.)}(\eta_k) = \Re\left(e^{-j\theta}\rho^{(.)}(\alpha)\right) \qquad \text{III.15}$$

4. On fixeθ, *N* et m pour mesurer la distorsion par:

La méthode D_{KL} suivant les lignes (X):

$$\hat{K}_{\theta,x} = \sqrt{N}.\frac{1}{m+1}\sum_{k=0}^{m}\left(K\left[\frac{p_{\theta,x,k}^{(1)}}{p_{\theta,x,k}^{(2)}}\right] + K\left[\frac{p_{\theta,x,k}^{(2)}}{p_{\theta,x,k}^{(1)}}\right]\right) \qquad \text{III.16}$$

Où $p_{\theta,x,k}^{(.)} = \gamma_{\theta,x}^{(.)}(\eta_1) + 1$ pour *k=0*

et: $p_{\theta,x,m}^{(.)} = 1 - \gamma_{\theta,x}^{(.)}(\eta_m)$ pour *k=m*

et: $p_{\theta,x,k}^{(.)} = \gamma_{\theta,x}^{(.)}(\eta_{k+1}) - \gamma_{\theta,x}^{(.)}(\eta_k)$ pour: *k=1,…,m-1*

La méthode D_{LL2} suivant les lignes(X) :

$$\gamma_{\theta,x}^2 = \sqrt{N}\,\frac{1}{m}\sum_{k=1}^{m}\left|\gamma_{\theta,x}^{(1)}(\eta_k) - \gamma_{\theta,x}^{(2)}(\eta_k)\right|^2 \qquad \text{III.17}$$

Avec: $\gamma_{\theta,x}^{(1)}(\eta_k)$, $\gamma_{\theta,x}^{(2)}(\eta_k)$ sont les deux signaux filtrés.

5. On filtre l'image **I(x,y)** suivant les colonnes(Y):

$$Y_y^{(.)} = \overline{\alpha}.Y_{y-1}^{(.)} + X_y^{(.)} \qquad \text{III.18}$$

6. L'Autocorrection suivant les colonnes(Y) :

$$\rho(\alpha) = \frac{E\left(Y_{y+1}(\alpha)\cdot \overline{Y}_y(\alpha)\right)}{E\left(\left|Y_y(\alpha)\right|^2\right)} \qquad \text{III.19}$$

et en suite, on prend:

$$Y_\theta^{2(.)}(\eta_k) = \Re\left(e^{-j\theta}\rho^{(.)}(\alpha)\right) \qquad \text{III.20}$$

7. On fixe θ, *N* et *m* pour mesurer la distorsion par :

La méthode D_{KL} suivant les colonnes(Y) :

$$\hat{K}_{\theta,y} = \sqrt{N}.\frac{1}{m+1}\sum_{k=0}^{m}\left(K\left[\frac{p^{(1)}_{\theta,y,k}}{p^{(2)}_{\theta,y,k}}\right] + K\left[\frac{p^{(2)}_{\theta,y,k}}{p^{(1)}_{\theta,y,k}}\right]\right) \qquad \text{III.21}$$

Où: $p^{(.)}_{\theta,y,k} = \gamma^{(.)}_{\theta,y}(\eta_1) + 1$ for $k=0$

et : $p^{(.)}_{\theta,y,m} = 1 - \gamma^{(.)}_{\theta,y}(\eta_m)$ for $k=m$

et : $p^{(.)}_{\theta,y,k} = \gamma^{(.)}_{\theta,y}(\eta_{k+1}) - \gamma^{(.)}_{\theta,y}(\eta_k)$ for: $k=1,\ldots,m-1$

La méthode D_{LL2} suivant les colonnes (Y) :

$$\gamma^2_{\theta,y} = \sqrt{N}\,\frac{1}{m}\sum_{k=1}^{m}\left|\gamma^{(1)}_{\theta,y}(\eta_k) - \gamma^{(2)}_{\theta,y}(\eta_k)\right|^2 \qquad \text{III.22}$$

Avec:

$\gamma^{(1)}_{\theta,y}(\eta_k)$, $\gamma^{(2)}_{\theta,y}(\eta_k)$ sont les deux signaux filtres

8. Finalement, on calcule la moyenne des deux mesures de distorsions suivant X et Y :

- La method D_{KL}: $\hat{K}_\theta = (\hat{K}_{\theta,x} + \hat{K}_{\theta,y})/2$
- La method D_{LL}: $\gamma^2_\theta = \sqrt{\left|\gamma^2_{\theta,x}\right| + \left|\gamma^2_{\theta,y}\right|}$

On obtient les maximums suivants (X) et (Y) de l'image pour avoir le contour voulu.

III .5. Résultats pratiques :[10]

En s'inspirant des travaux réalisés par Ta-Hsin sur la segmentation audio et en tenant compte de nos travaux expérimentaux nous montrons ici quelque uns des résultats obtenus. La Figure III.2 représente une image de dimension 256x256 codée en 256 niveaux de gris, représentant différents objets sur laquelle on teste l'algorithme de filtrage paramétrique.

En ce qui concerne la taille des fenêtres Hamming, nos testes sont effectués sur des fenêtres de taille **N=4.**

La figure(**III.2**) suivante montre le résultat de détection de contours par la méthode de filtrage paramétrique à base des méthodes D_{KL} et D_{LL}.

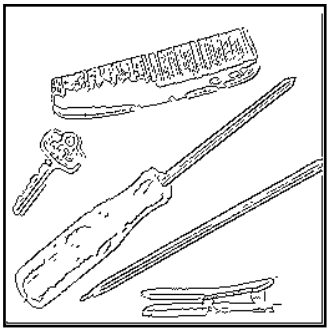
La figure(**III.2**.**A**), montre l'image originale d'objets.

La figure (**III.2**.**B**) montre l'image contour quand la mesure de distorsion DKL est utilisée avec les paramètres suivants : $\eta_a = 0.0$, $\eta_b = 0.9$, $\phi = 0.00063\pi$, $m = 2$

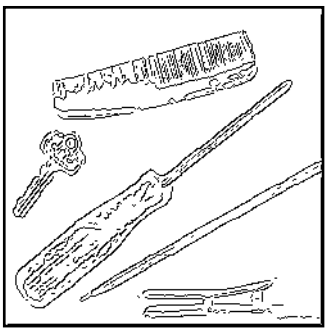
La figure (**III.2**.**C**) montre l'image contour quand la mesurée distorsion DLL est utilisée avec les paramètres suivant : $\eta_a = 0.5$, $\eta_b = 0.5$, $\phi = 0.05\pi$, $m = 2$



A) image originale d'objets**.**



B) image contour avec la méthode DKL



C) image contour avec la méthode DLL2

Figure III.2 : résultat de détection de contours par la méthode de filtrage paramétrique à base des méthodes D_{KL} et D_{LL}.

DKL : *distance* de Kullback-Leibler

III.6 Conclusion :

Dans ce chapitre, on a présenté la nouvelle méthode de détection de contour d'image et de caractérisation de l'image. La méthode PF emploie un filtre judicieusement déterminé, qui préserve la structure de corrélation du signal en entrée dans l'autocorrélation $\gamma_\theta(\eta)$ de la sortie. Car l'application des méthodes de détermination de distorsion D_{KL} et D_{LL2}, permet la détermination des contours de l'image.
Dans le chapitre qui suit, nous allons présenter les critères d'évaluation de cette méthode PF de détection des contours d'image.

Chapitre IV : Évaluation de la segmentation d'images

IV.1 Introduction :

La segmentation d'une image est une étape essentielle en traitement d'images dans la mesure où elle conditionne son interprétation. Il existe une multitude de méthodes de segmentation dont l'efficacité reste difficile à évaluer.

Les méthodes de mesure de la qualité de la segmentation qui existent actuellement peuvent être regroupées en trois grandes catégories :

- évaluation par un expert humain (visuelle)
- évaluation avec référence (supervisée) ou l'on fait appel à des images de synthèses construites par un expert du domaine d'application
- évaluation sans référence (non supervisée) dans le cas ou l'on ne dispose de la vérité terrain et qu'il n'y a pas suffisamment d'informations à priori sur l'image.

L'objectif principal d'une méthode d'évaluation est d'être capable de comparer de différents résultats de segmentation afin de trouver la meilleure segmentation.

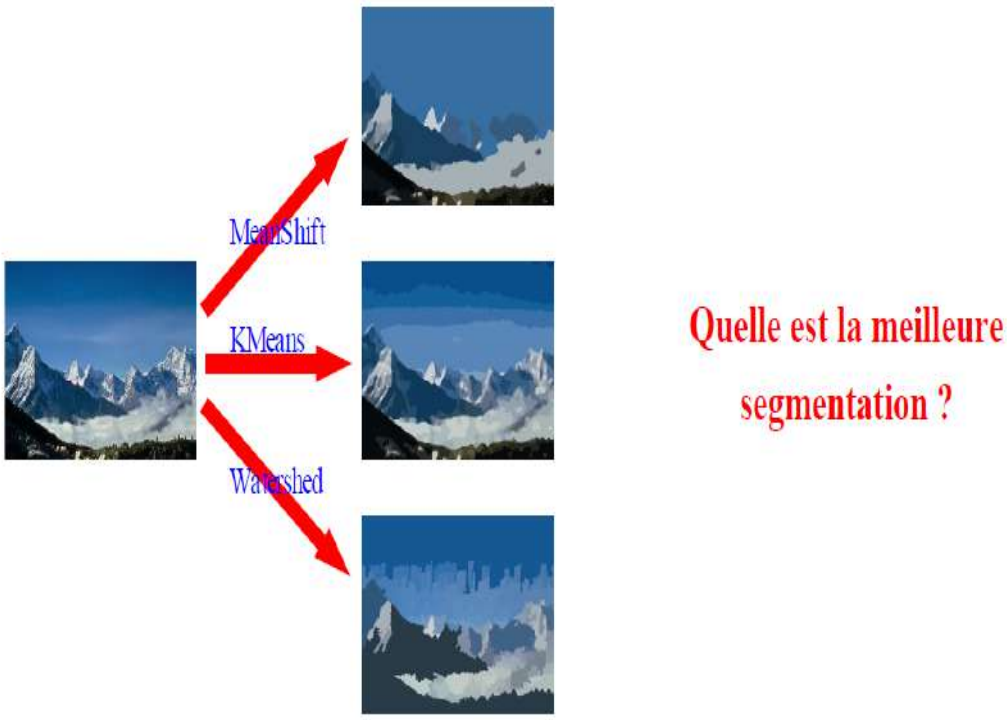


Figure IV.1 : Quelques méthodes de segmentation

IV. 2 Les différents types d'erreurs de segmentation :[14]

IV. 2.1 Erreur de sur- segmentation :

Les algorithmes de segmentation peuvent affecter à tord des pixels à un contour ou à une région, alors que ceux-ci n'appartiennent pas à ce contour ou a cette région.

IV. 2.2 Erreur de sous- segmentation :

Lorsque la différence de niveau de gris (ou de couleur) entre de région voisines n'est pas très marquées, un certain nombre d'algorithmes de segmentation nr détectent pas le contour ou la région, donc certain pixels appartenant à un contour ou à une région ne seront pas pris comme tels.

IV. 2.3 Erreur de localisation :

En détection de contour, le contour est défini par convention quand deux régions homogènes se touchent, comme la limite ou la ligne de séparation d'un pixel entre ces deux régions .les pixels contours lors de la segmentation doivent être placés dans l'une ou l'autre des régions .deux méthodes différents de segmentation peuvent parfaitement localiser la limite désirée dans une image, mais pendant que l'une dessine le contour du côté sombre, l'autre le placera du côté plus clair.

Les images contours obtenues sont évidemment différents et les pixels contour sont considérés comme mal classés, donc ils ne sont pas à leur place réelle et donnent des erreurs de localisation. Pour la segmentation en régions nous savons que la plupart des algorithmes de croissance de régions ont du mal à s'arrêter exactement à la frontière des objets, donc les pixels d'une région peuvent être attribués à tord à la région voisine et donneront ainsi des erreurs de localisation.

IV. 3 Évaluation de la segmentation :

IV. 3.1 Évaluation visuelle:

Une méthode très simple et usuelle est d'évaluer des résultats de segmentation avec un grand nombre de juges. Pour chaque paires d'images, les juges doivent choisir le résultat qu'ils considèrent être meilleur. Cette évaluation est très subjective car elle dépend des critères utilisés par l'expert humain. Ces critères vont dépendre du domaine d'intérêt de l'expert, du but précis qu'il cherche à atteindre, de plus celui-ci a souvent une idée globale de l'image et approximative de ce qui doit être détecté.

L'avantage principal des méthodes d'évaluation visuelle est de permettre d'évaluer et de classifier des algorithmes de segmentation lorsqu'aucune donnée sur des objets à segmenter n'est disponible. Le défaut principal de ces méthodes est l'impossibilité de les automatiser.

IV. 3.2 Évaluation supervisée :

Le principe des critères supervisés est de tirer parti d'une connaissance supplémentaire pour effectuer le jugement sur la qualité d'un résultat issu d'un algorithme de traitement d'images.

Cette connaissance est en général une référence représentant un résultat idéal de segmentation que l'on nomme vérité terrain (*ground truth*). Nous pouvons obtenir une vérité terrain à partir d'une image synthétique ou d'une image naturelle.

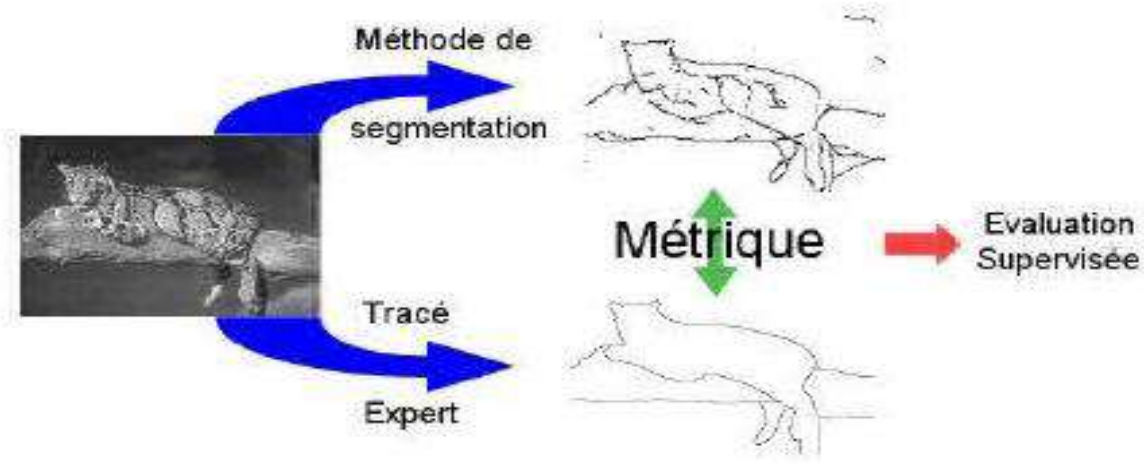


Figure IV.2 : Évaluation supervisée.

IV. 3.2.1 Évaluation avec vérité terrain experte :

Une vérité terrain experte est un résultat de segmentation d'une image naturelle tracé manuellement par des experts. C'est souvent difficile d'obtenir une bonne vérité terrain experte à cause de la complexité des images naturelles. Il y a toujours une incertitude sur la précision des donnés acquises.

Une méthode simple abordée est d'utiliser des matrices de confusion pour l'évaluation quantitative le résultat de segmentation d'une image contenant des cellules. En compter le nombre de pixels corrects et faux dans l'image résultat par rapport à la vérité terrain, nous pouvons calculer le pourcentage de pixels corrects et celui de pixels bien détectés. Un critère similaire proposé par Shufelt permet de mesurer la qualité de détection de bâtiments. Sans utiliser la matrice de confusion, on classe les pixels de l'image segmentée en certaines catégories : les vrais positifs, les vrais négatifs, les faux positifs, les faux négatifs. À partir de ces catégories, quelques mesures sont alors définies pour évaluer le résultat de segmentation.

L'avantage principal de ces critères est de permettre d'évaluer des algorithmes de segmentation avec des vérités terrains qui sont obtenues à partir des images naturelles. L'inconvénient majeur de ces méthodes est que c'est difficile de définir une bonne vérité terrain lorsque chaque expert a sa propre opinion.

IV. 3.2.2 Évaluation avec vérité terrain synthétique :

Soit une image synthétique créée par un programme. Cette image est souvent beaucoup plus simple qu'une image naturelle. Nous avons toute information de cette image, donc nous sommes capables de créer une bonne vérité terrain correspondante.

Il y a plusieurs critères que nous pouvons utiliser pour mesurer la différence entre un résultat et la vérité terrain synthétique correspondante, et ces critères sont partitionnés en deux groupes : les critères évaluant un résultat de détection de frontières et ceux évaluant un résultat de segmentation en régions.

IV. 3.2.2.1 Evaluation de la segmentation en contours :

L'une des idées les plus anciennes pour mesurer la qualité d'une segmentation contour est de calculer l'écart entre les contours obtenus et les théoriques (sur/sous-détection et erreur de localisation). Ainsi, pour pouvoir effectuer une mesure d'évaluation de segmentation contour l'idée est d'utiliser un type d'erreur de détection ou une combinaison des trois. Ces trois mesures d'erreurs donnent des indices globaux sur la qualité d'un résultat. Nous considérons qu'un résultat de détection est bon si ces trois erreurs sont petites.

1- Critère d'erreur de sur détection :

$$EEE_{SUR}(I_F, I_{Ref}) = \frac{card\,(I_F) - card\,(I_f \cap I_{Ref})}{card\,(I) - card\,(I_{Ref})} \qquad \text{IV.1}$$

2- Critère de mesure L'erreur de sous-détection sous segmentation :

$$ERR_{SOUS}(I_f, I_{ref}) = \frac{card\,(I_{Ref/F}^{Cont})}{card\,(I_{Ref}^{Cont})} \qquad \text{IV.2}$$

3- Critère de mesure d'erreur de localisation :

$$ERR_{LOC}(I_F, I_{Ref}) = \frac{card\,(I_{Ref/F} \cup I_{F/Ref})}{card\,(I)} \qquad \text{IV.3}$$

4- Critère Lq :

En utilisant ce critère, nous pouvons calculer la divergence selon l'erreur quadratique moyenne de l'image segmentée par rapport à la vérité terrain.

$$D_{L(q)}(I_1, I_2) = \left(\frac{1}{card\,(X)} \sum_{(i,j)\in X} \left| g_{I_1}(i,j) - g_{I_2}(i,j) \right|^q\right)^{\frac{1}{q}} \qquad \text{IV.4}$$

5- Critère de Bhattacharyya :

$$D_{BHA}(I_1, I_2) = -log(\frac{1}{card\ (X)} \sum_{(i,j)\in X} \sqrt{g_{I_1}(i,j) \times g_{I_2}(i,j)}\ ) \quad \text{IV.5}$$

6- Critère de Jensen :

$$D_{GEN}(I_1, I_2) = J(\frac{I_1+I_2}{2}, I_1) \quad \text{IV.6}$$

Où

$$j(I_1, I_2) = H_\propto\left(\sqrt{I_1 \times I_2}\right) - \frac{H_\propto(I_1)+H_\propto(I_2)}{2} \quad \text{IV.7}$$

Avec

$$H_\propto(I_1) = \frac{1}{1-\propto} log_2\left(\sum_{(i,j)\in I_1}(g_{I_1}(i,j))^\propto\right), 3 < \propto \in N \quad \text{IV.8}$$

7- Critère de Kullback :

$$D_{KUL} = \frac{1}{card\ (X)} \sum_{(i,j)\in X} \left(g_{I_1}(i,j) - g_{I_2}(i,j)\right) log(\frac{g_{I_1}(i,j)}{g_{I_2}(i,j)}) \quad \text{IV.9}$$

8- Critère de pratt :

$$PRA\left(I_F, I_{Ref}\right) = \frac{1}{MP} \sum_{K=1}^{card\ (I_F^{cont})} \frac{1}{1+\propto d^2(I_F^{Cont}\ (k), I_{Ref}^{Cont})} \quad \text{IV.10}$$

Où

$$MP = max\left\{card\left(I_{Ref}^{Cont}\right), card(I_F^{Cont})\right\}, 0 \leq PRA(I_F, I_{Ref}) \leq 1. \quad \text{IV.11}$$

9- Critère de Haussdorf :

$$D_{HAU}\left(I_F, I_{Ref}\right) = max(h\left(I_F, I_{Ref}\right), h\left(I_{Ref}, I_F\right)) \quad \text{IV.12}$$

Où

$$h(I_1, I_2) = max_{a\in I_F^{Cont}}\ min_{b\in I_{Ref}^{Cont}}\ \|a-b\| \quad \text{IV.13}$$

IV. 3.2.2.2 Évaluation de segmentation en régions :[14]

1- Critère de Yasnoff :

Yasnoff a proposé une série de mesures d'erreur de classification lorsque les classes sont supposées connues (étant donnée une segmentation, il faut connaître pour chaque région celle qui lui est associée dans une autre segmentation). Les erreurs sont alors calculées sur la base d'une matrice de confusion.

2-Critère de Yasnoff 1 :

$$ERR_{YAS}^{1}(K) = 100 \times \frac{\sum_{i=1}^{n} M_{iK} - M_{KK}}{\sum_{i=1}^{n} M_{iK}} \qquad \text{IV.14}$$

3-Critère de Yasnoff 2 :

$$ERR_{YAS}^{2}(K) = 100 \times \frac{\sum_{i=1}^{n} M_{ik} - M_{KK}}{\sum_{j=i}^{n} \sum_{i=1}^{n} M_{ij} - \sum_{i=1}^{n} M_{iK}} \qquad \text{IV.15}$$

4-Critère de Yasnoff 3 :

$$ERR_{YAS}^{3}(I_R, I_{Ref}) = \frac{100}{card\,(I_R)} \times \sqrt{\sum_{a \in I_R, a \notin R_a} min_{b \in R_a} d(a,b)} \qquad \text{IV.16}$$

avec :

M_{ij} : le nombre de pixels mal classés

M_{ii} *:* le nombre de pixels bien classés

n **:** le nombre de région dans la segmentation idéale

5-Critère d'évaluation de Vinet :

La distance de Vinet permet de calculer une mesure de dis similarité entre deux classifications en calculant le recouvrement maximal entre les classes de deux résultats de segmentation (une classe du résultat et une classe de la vérité terrain). Cette méthode est facile à mettre en œuvre. Cependant, elle n'apparie pas toutes les classes, donc elle ne prend pas en compte toute l'information (entre autre la dispersion spatiale des pixels). Elle part de l'hypothèse que deux classes sont à apparier si elles ont un ensemble maximal commun de pixels Cette hypothèse est restrictive et privilégie les grandes régions.

Le **critère de Vinet** est :

$$VIN(I_L, I_{Ref}) = card(I) - \sum_{c'} card(L_i \cap L_j^{Ref}) \qquad \text{IV.17}$$

IV. 3.3 Évaluation non supervisée :

En l'absence de connaissances sur les images segmentées, Ce sont les critères d'évaluation non supervisée qui nous permettent de créer un système d'évaluation automatique.ces critères permettant d'estimer la qualité de résultat de segmentation à partir de statistiques calculées sur chaque région ou contour détecté.

IV. 3.3 .1Évaluation de la segmentation en contours :[13] [14]

1-Mesure de Kitchen et Rosenfeld :

Kitchen et Rosenfeld ont proposé un critère d'évaluation de cartes de frontières fondée sur la cohérence locale des frontières. ***Elle utilise deux critères de caractérisation des contours qui sont la continuité et la minceur***. Les frontières sont supposées être générées par un détecteur de frontières de type gradient, suivi par un seuillage.

Des critères définis par Levine et Nazif permettent d'évaluer deux types de frontières : les frontières séparant deux objets de la scène et les lignes qui traversent un objet Puis, nous devons définir une caractéristique ou un attribut (par exemple : le niveau de gris, un calcul d'attribut de texture) dont on calcule la moyenne sur une région. Enfin, nous calculons le contraste de part et d'autre de chaque frontière et le gradient moyen le long de chaque ligne. De plus, Levine et Nazif définissent un indice de connexité des frontières pour mesurer les frontières manquantes selon le nombre de pixels et d'extrémités de chaque ligne.

2-Critères de Levine et Nazif :

$$LEV5(l_i) = \frac{|C_g(l_i) - C_d(l_i)|}{C_g(l_i) + C_d(l_i)} \qquad \text{IV.18}$$

avec

$$C_g(l_i) = \sum_{R_k \in W_g(l_i)} card(l_i \cap R_K)\overline{a}\ (R_K) \qquad \text{IV.19}$$

IV. 3.3 .2Évaluation de la segmentation en région :

1-Critère de Borsotti :

$$BOR(I_R) = \frac{\sqrt{NR}}{10^4 \times card\ (I)} \sum_{K=1}^{NR} \left[\frac{e_K^2}{1 + \log(card\ (R_K))} + \left(\frac{x(card\ R_K))}{card\ (R_K)} \right)^2 \right] \qquad \text{IV.20}$$

χ (*card* (*Rk*)) correspond au nombre de région ayant une aire égale a *card*(*Rk*).

L'idée principale de Borsotti est :

- les régions non homogènes (les régions deviennent moins homogènes quand le nombre de régions diminue).
- les régions dont l'aire est égale a beaucoup d'autres régions.

L'entropie pour une région *Rk* est définie comme suit :

$$H(R_K) = -\sum_{m \in V_K} \frac{L_K(m)}{S_K} log \frac{L_K(m)}{S_K} \quad \text{IV.21}$$

Vk : correspond a l'ensemble des valeurs possibles des intensités dans la région *Rk*

Lk(*m*) : correspond au nombre de pixels dans la région *Rk* dont l'intensité est égale a *m*.

L'entropie de région désirée (*excepte régions entropie*) de l'image segmentée *IR* est la suivante :

$$H_r(I_R) = \sum_{K=1}^{NR} \frac{card\,(R_K)}{card\,(I_R)} H(R_K) \quad \text{IV.22}$$

Huizhang a propose l'entropie de disposition (qui est définie comme suit :

$$H_r(I_R) = -\sum_{K=1}^{NR} \frac{card\,(R_K)}{card\,(I_R)} log \frac{card\,(R_K)}{card\,(I_R)} \quad \text{IV.23}$$

Le critère de Huizhang est donc la somme de deux entropies *Hr*(*IR*) et *Hl*(*IR*).

2-Le critère de Levine et Nazif (la première version)

Considèrent la variation de niveau de gris de chaque région comme la mesure de qualité. Sur une image faiblement ou non texture.

$$LEV1(I_R) = 1 - \sum_{K=1}^{NR} \frac{\sum_{(i,j)\in R_k} \left[g_1(i,j) - \sum_{(m,n)\in R_k} g_1(m,n)\right]^2}{card\,(I) \times (max_{(i,j)\in R_K}(g_1(i,j)) - min_{(i,j)\in R_K}(g_1(i,j)))^2} \quad \text{IV.24}$$

Où

$$LEV2(I_R) = \sum_{K=1}^{NR} \sum_{(i,j)\in R_k} \left[g_1(i,j) - \frac{1}{card\,(R_K)} \sum_{(m,n)\in R_k} g_1(m,n)\right]^2 \quad \text{IV.25}$$

3-Les critères de Levine et Nazif (la troisième version) :

Considère le contraste de chaque région et de sa région adjacente comme la mesure à évaluer :

$$LEV3(I_R) = \frac{\sum_{K=1}^{NR} w_{R_K} C2(R_k)}{\sum_{K=1}^{NR} w_{R_k}} \quad \text{IV.26}$$

$$C1(R_a, R_b) = \frac{|\bar{g}_1(R_a) - \bar{g}_1(R_b)|}{\bar{g}_1(R_a) + \bar{g}_1(R_b)} \quad , \quad C2(R_a) = \sum_{R_K \in W(R_a)} P_{R_a R_K}\, C1(R_a, R_K) \quad \text{IV.27}$$

et

$$w_{R_j} = \frac{1}{\sqrt{2\pi}\,\sigma e^{\frac{(card\,(R_j)-\mu)^2}{\sigma^2}}} \qquad \text{IV.28}$$

4-Les critères de Zeboudj :

Il donne un bon résultat de segmentation en régions doit avoir de grandes valeurs de contraste inter-région et d'homogénéité intra-région.

Le contraste entre deux pixels *x* et *y* de l'image I est defini comme suit :

$$C_I(x,y) = \frac{|g_1(x)-g_1(y)|}{max_{\,z\in I}(g_1(z))-1} \qquad \text{IV.29}$$

Le contraste intérieur d'une région *Rk* de l'image *I* est défini comme suit :

$$C_{Int}(R_K) = \frac{1}{card\,(R_K)}\sum_{x\in R_k} max\,\{c_I(x,y), y \in w(x) \cap R_k\} \qquad \text{IV.30}$$

W(*x*) :l'ensemble des pixels adjacents a *x*

Le contraste extérieur d'une région *Rk* de l'image *I* est défini comme suit :

$$C_{Ext}(R_k) = \frac{1}{card\,(F_{R_K})}\sum_{x\in F_{R_K}} max\{C_I(x,y), y \in W(x), y \notin R_K\} \qquad \text{IV.31}$$

FRk :la frontière de la région *Rk*

Le contraste global de la région *Rk* de l'image originale *I* est calcule comme suit :

$$C(R_K) = \begin{cases} 1 - \frac{C_{Int}\,(R_K)}{C_{Ext}\,(R_K)} \; si\; 0 < C_{Int}(R_K) < C_{Ent}(R_K) \\ C_{Ext}(R_K)\; si\; C_{Int}(R_K) = 0 \\ 0 \quad sinon \end{cases} \qquad \text{IV.32}$$

Le contraste global de l'image *I* est exprime comme le suivant :

$$ZEB(I_R) = \frac{1}{card\,(I)}\sum_{K=1}^{NR} card(R_k) \times C(R_K) \qquad \text{IV.34}$$

La valeur des critères de Zeboudj est entre 0 et 1. Plus cette valeur est grande (c'est à dire qu'elle est plus proche de 1), meilleur est suppose être le résultat de segmentation.

IV. 4 Conclusion :

L'avantage principal des critères d'évaluation non supervisée est qu'ils requièrent seulement une métrique de mesure de qualité. Alors, ils sont automatisable et ne nécessitent aucun expert. L'inconvénient majeur de ces critères est l'incertitude sur la fonction de calcul de caractéristiques.

Dans notre mémoire on applique les critères d'évaluation en contour pour évaluer la méthode de segmentation par filtrage paramétrique et la compare avec la méthode de Canny.

Chapitre V : Implémentation logiciel et évaluation des résultats

V .1Introduction :

Ce chapitre va être le fruit de tous ce qui a été fait au paravent. Nous allons présenter les résultats d'implémentation des méthodes de détection de contours d'images de filtrage paramétrique (PF) et la méthode de Canny d'une part et d'autre part de présenter les résultats d'évaluation des ces méthodes en utilisant quelques méthodes d'évaluation.

L'implémentation des algorithmes a été réalisée en utilisant le logiciel MATLAB version 7.8.0.347 (R2009a).

V .2 mise en œuvre des algorithmes de détection de contour :

V .2.1 algorithme de filtrage paramétrique :

V .2.1.1 L'algorithme de filtrage paramétrique D_{KL} :

Cet algorithme est constitué des étapes suivantes :

1. Lecture et affichage de l'image originale X(i,j).

2. Prendre les fenêtres de Hemming à partir de signal image original : $X_t^{(1)}, X_t^{(2)}$

3. Filtrage Paramétrique H(z^{-1}, a). $Y^{(.)}(i,j) = \overline{\propto} . Y(i,j-1) + X_i^{(.)}(i,j)$.

4. Calcul L'autocorrection de premier ordre : $$\rho(\alpha) = \frac{E(Y_{y+1}(\alpha).\bar{Y}_y(\alpha))}{E\left(\left|Y_y(\alpha)\right|^2\right)}$$

5. prendre La partie réelle : $Y_\theta^{2(.)}(\eta_k) = \Re(e^{-j\theta}\rho^{(.)}(\alpha))$.

6. Mesure de distorsion par la méthode de :D_{KL}

7. affichage des résultats.

V .2.1.2 L'algorithme de filtrage paramétrique D_{LL} :

Dans ce cas, on suit les mêmes étapes de l'algorithme précédent et on remplace la méthode de D_{KL} par la méthode de D_{LL} pour la mesure de distorsion

1. Lecture et affichage de l'image originale X(i,j).

2. Prendre les fenêtres de Hemming à partir de signal image original : $X_t^{(1)}, X_t^{(2)}$

3. Filtrage Paramétrique H(z^{-1}, a). $Y^{(.)}(i,j) = \overline{\propto} . Y(i,j-1) + X_i^{(.)}(i,j)$.

4. Calcul L'autocorrection de premier ordre : $$\rho(\alpha) = \frac{E(Y_{y+1}(\alpha).\bar{Y}_y(\alpha))}{E\left(\left|Y_y(\alpha)\right|^2\right)}$$

5.prendre La partie réelle : $Y_\theta^{2(.)}(\eta_k) = \Re(e^{-j\theta}\rho^{(.)}(\alpha))$.

6. Mesure de distorsion par la méthode de :D_{LL}

7. affichage des résultats

V .2.1. 3 Présentation des résultats :

Dans ce qui suit, on présente les résultats de détection des contours par les deux algorithmes de filtrage paramétrique en utilisant plusieurs images.

- **Pour l'image <<woman.png>> :**



Image originale (256x256)



Image segmenté par PF_DLL



Image segmenté par PF_DKL

- **pour l'image <<imori.bmp>> :**

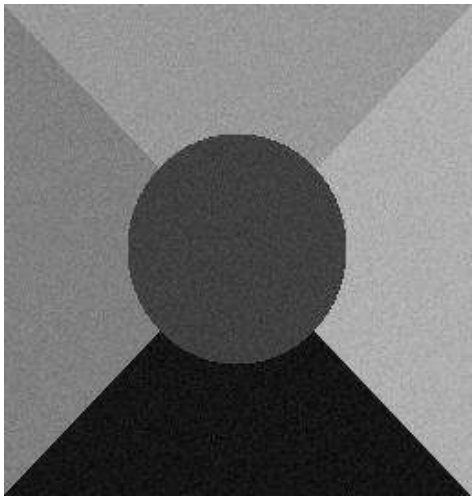


Image originale



Image segmenté par PF_DLL



Image segmenté par PF_DKL

- **pour l'image <<etoile.bmp>> :**

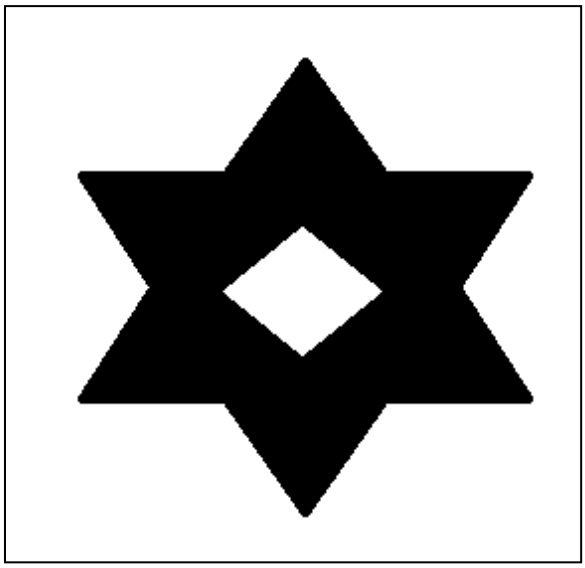


Image originale

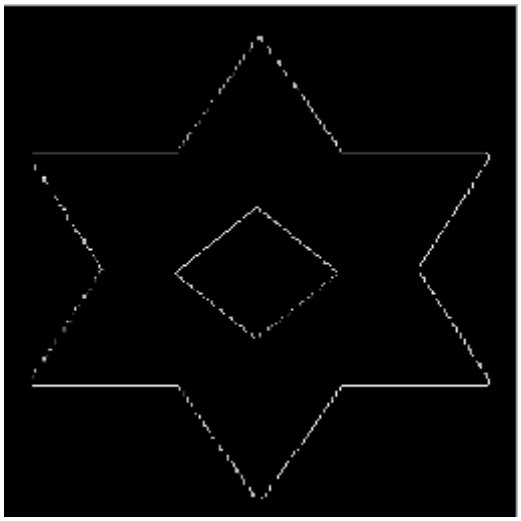


Image segmenté par PF_DLL



Image segmenté par PF_DKL

- **pour l'image <<formes.bmp>> :**

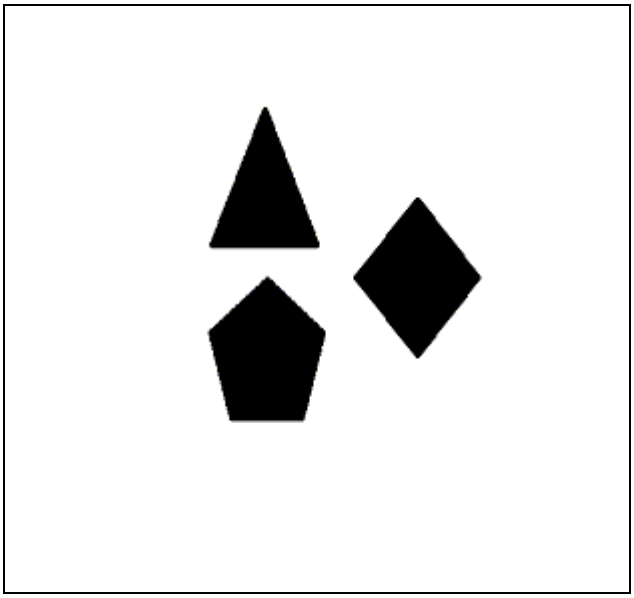


Image originale (256x256)

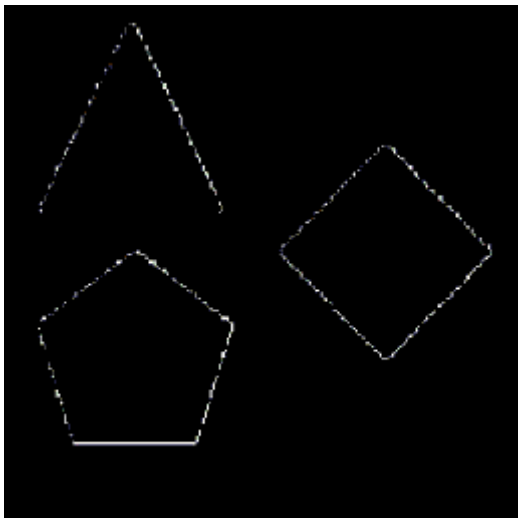


Image segmenté par PF_DLL



Image segmenté par PF_DKL

V .2.2 l'algorithme de canny :

V .2.2.1 présentation de l'algorithme de canny :

Cet algorithme est constitué des étapes suivantes :

1. Lecture et affichage de l'image originale (NxN).
2. Choix de l'ondelette et l'échelle à utiliser.
3. Conversion de l'ondelette en un masque de taille (SxS).
4. Convolution entre le masque d'ondelette et l'image.
5. Calcul des maximas au sens de Canny (contours) de l'image, par le test de l'énergie moyenne calculée.
6. Chainage des maximas au sens de Canny suivant les échelles choisis.
7. Afichage des resultats.

V .2.2. 2 présentation des résultats :

- **pour l'image <<woman.png>> :**



Image originale



Image segmenté par canny

- **pour l'image <<imori.bmp>> :**

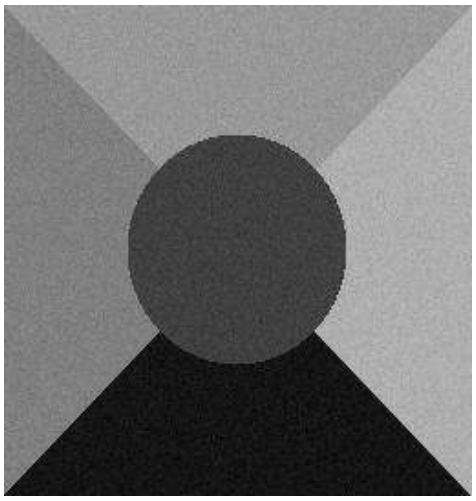


Image originale

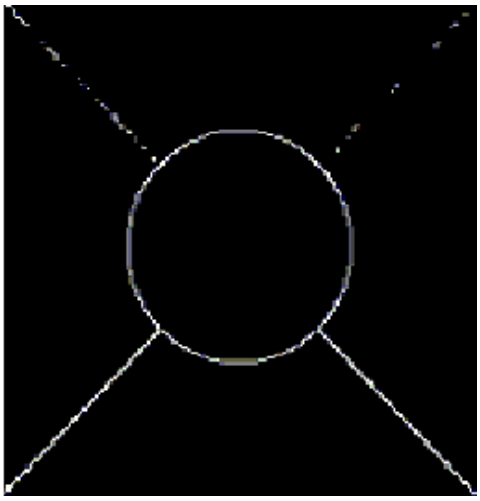


Image segmenté par canny

- **pour l'image <<etoile.bmp>> :**

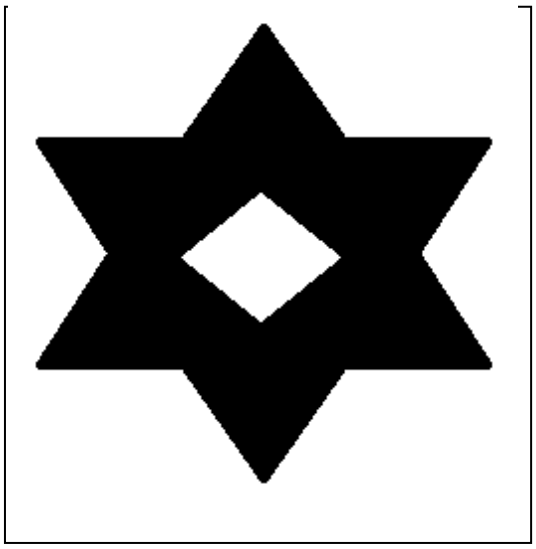


Image originale

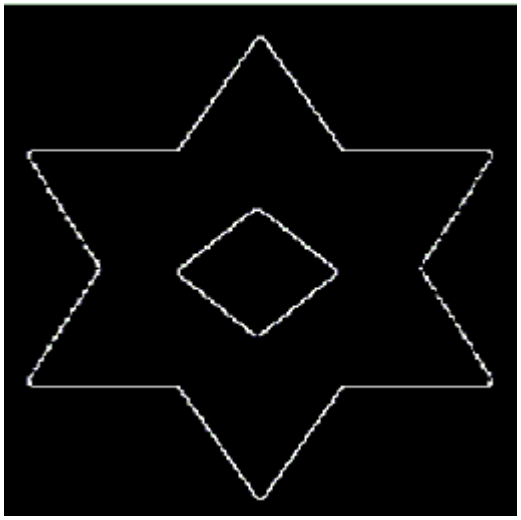


Image segmenté par canny

- **pour l'image <<formes.bmp>> :**

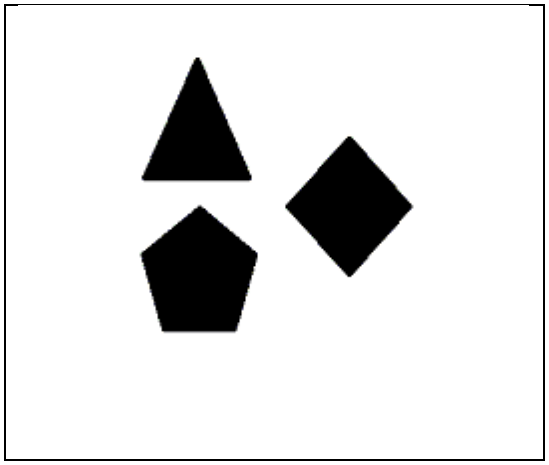


Image originale

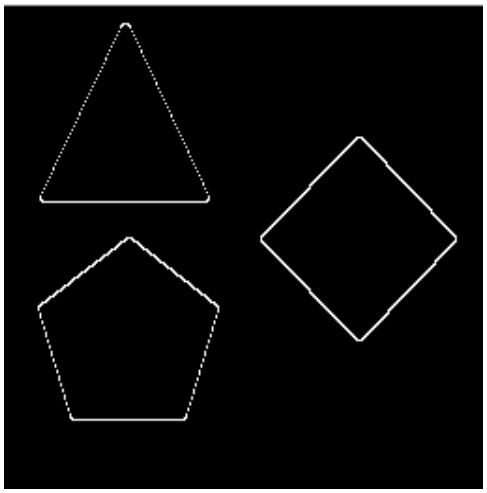


Image segmenté par canny

V .3 Évaluation des résultats de segmentation :

En ce qui concerne l'évaluation des résultats de segmentation, on se limite à la Méthodes visuelle et la méthode basée sur les critères d'évaluation supervisés.

V .3.1 méthode visuelle :

D'abord, nous organisons les résultats de segmentation de les trois algorithmes Précédents pour les déférentes images utilisées.



<< woman.png>>



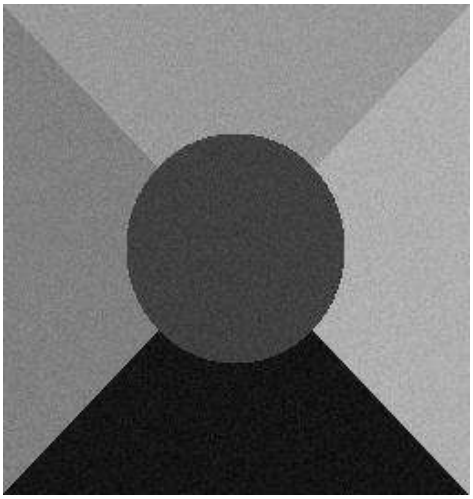
PF_DLL



PF_DKL



canny



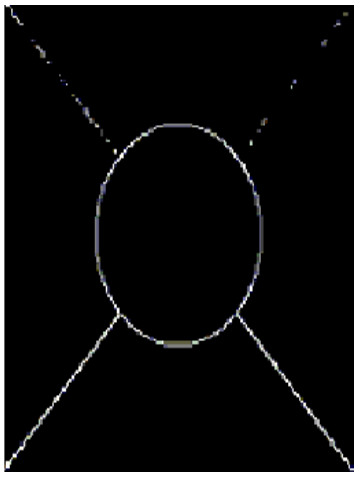
<<imori.bmp>>



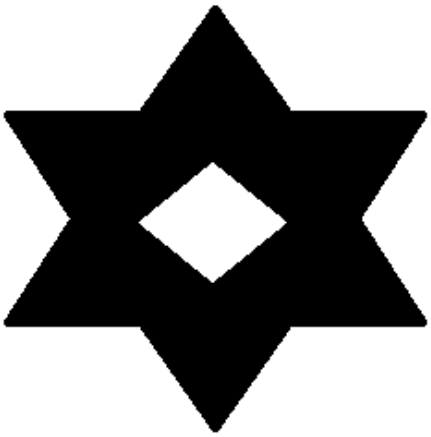
PF_DLL



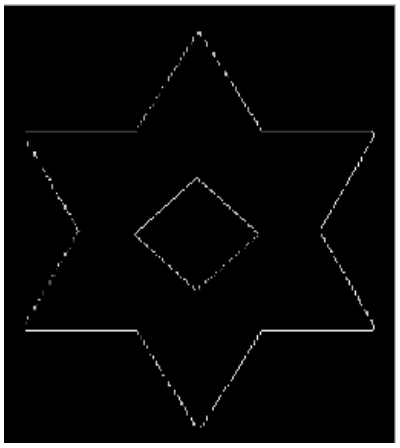
PF_DKL



canny



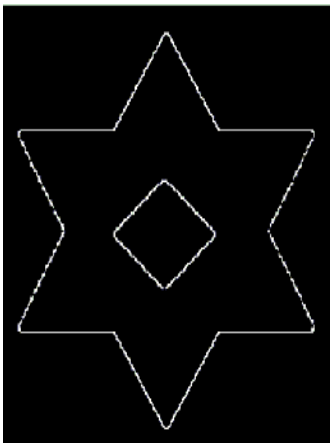
<<etoile.bmp>>



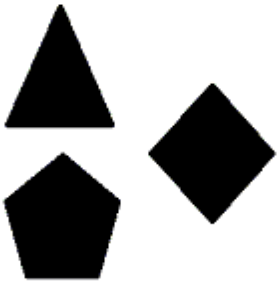
PF_DLL



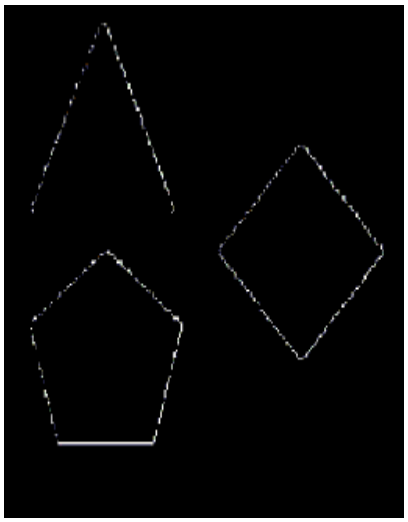
PF_DKL



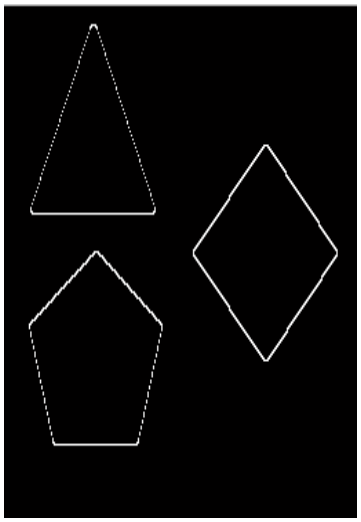
canny



<< formes.bmp>>


PF_DLL


PF_DKL


canny

D'après les observations visuelles, nous pouvons noter les remarques suivantes :

1- Les lignes des contours dans les résultats de segmentation basées sur la méthode de Canny continues, contrairement à la méthode PF D_{LL} qui est à son tour meilleure que la méthodes PF D_{KL} .
2- Les traits sont plus clairs dans les résultats basés sur la méthode de segmentation de Canny que dans celles de la méthode de filtrage paramétrique.

3- pas des courbures des contours dans les résultats de segmentation

Sur la base de ces observations réalisées à l'œil nu, on peut classer les résultats de Segmentation des différents images comme suit :

les résultats obtenus par la méthode de canny sont meilleurs par rapport aux ceux obtenus par la méthode PF.

V .3.2 la méthode d'évaluation supervisé :

V .3.2.1 présentation des critères d'évaluation :

Nom de critère	Équation de critère
erreur de sur détection	$EEE_{SUR}(I_F, I_{Ref}) = \frac{card(I_F) - card(I_f \cap I_{Ref})}{card(I) - card(I_{Ref})}$
mesure d'erreur de localisation	$ERR_{LOC}(I_F, I_{Ref}) = \frac{card(I_{Ref/F} \cup I_{F/Ref})}{card(I)}$

Critère Lq	$D_{L(q)}(I_1, I_2) = (\frac{1}{card(X)} \sum_{(i,j)\in X} \left\|g_{I_1}(i,j) - g_{I_2}(i,j)\right\|^q)^{\frac{1}{q}}$
Critère de Bhattacharyya	$D_{BHA}(I_1, I_2) = -log(\frac{1}{card(X)} \sum_{(i,j)\in X} \sqrt{g_{I_1}(i,j) \times g_{I_2}(i,j)})$
Critère de Kullback	$D_{KUL} = \frac{1}{card(X)} \sum_{(i,j)\in X} \left(g_{I_1}(i,j) - g_{I_2}(i,j)\right) log(\frac{g_{I_1}(i,j)}{g_{I_2}(i,j)})$

Tableau V.1 : Critères d'évaluation supervisés

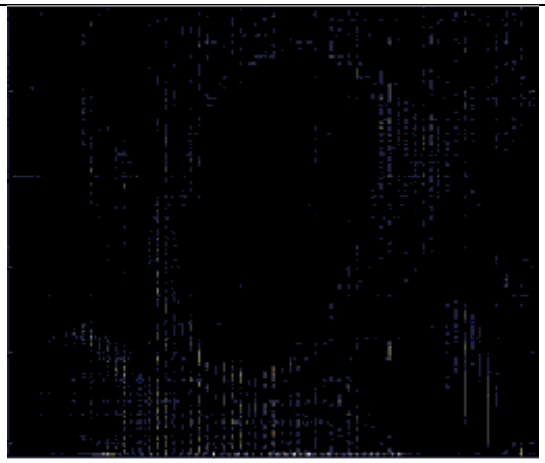
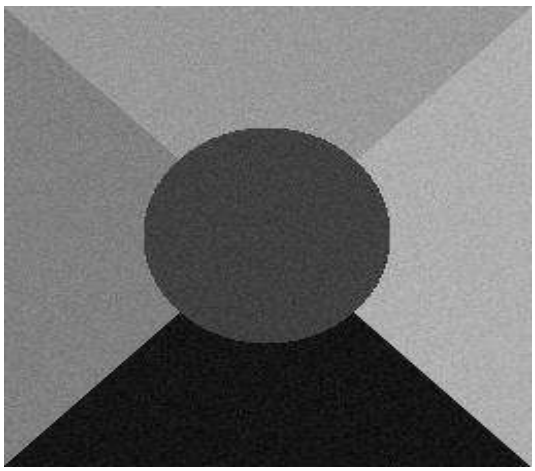
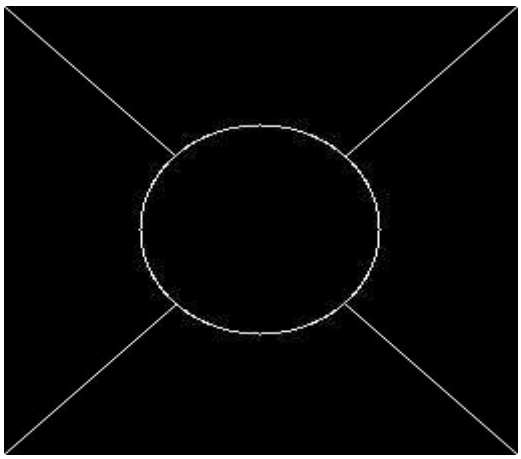
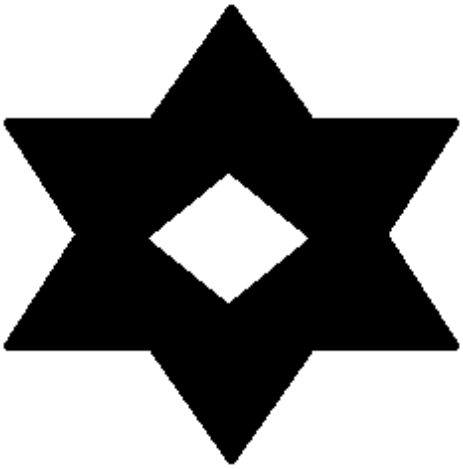
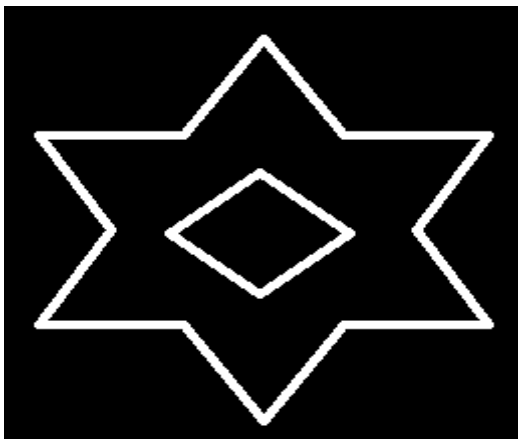
V .3.2.2 présentation des résultats d'évaluation :

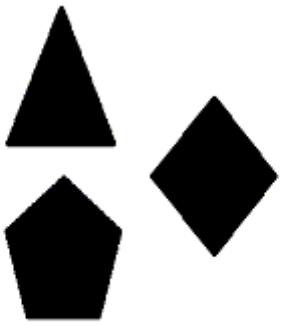
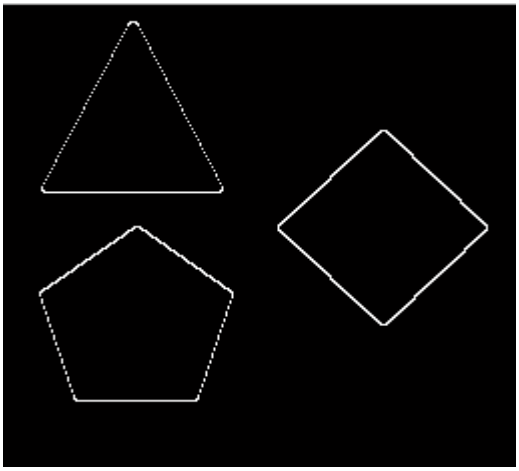
Pour cette évaluation, nous appliquons les critères cités en utilisant les images **<<imori.bmp>> , <<etoile.bmp>> , <<formes.bmp>>** où leurs vérités terrains sont obtenues à l'aide du logiciel ***paint***.

Dans le cas de l'image **<<woman.png>>,** nous avons utilisé le résultat de segmentation de la méthode de canny comme vérité terrain pour l'évaluation de la méthode de filtrage paramétrique , et le résultat de segmentation de la méthode de filtrage paramétrique comme vérité terrain pour l'évaluation de la méthode de canny .

.les vérités terrains :

Image originale	**Vérité terrain**
 **<<woman.png>>**	 **woman.canny**

	
<<woman.png>>	**woman.PF_DLL**
	
<<imori.bmp>>	**<<imori Vérité terrain >>**
	
<<etoile.bmp>>	**<<etoile Vérité terrain >>**

 **<<formes.bmp>>**	 **<<formes Vérité terrain >>**

- **Résultat d'évaluation de la méthode de Canny :**

critères	**<<woman.png>>**	**<<etoile.bmp>>**	**<<formes.bmp>>**	**<<imori.bmp>>**
erreur de sous détection	0.028415	0.003612	0.00526	0.00669
erreur de sur détection	0.0588	0.6388	0.6850	0.1678
mesure d'erreur de localisation	0.0304	0.0619	0.0591	0.010025
Critère Lq	52.24533	21.2880	19.0686	5.88832
Critère de Bhattacharyya	0.5669750	0.3256	0.29050	0.06047
Critère de Kullback	110.44817	48.2771	44.6336	12.5216

Tableau V.2 : Résultats d'évaluation de la segmentation de canny

- **Résultat d'évaluation de la méthode de filtrage paramétrique :**

critères	**<<woman.png>>**		**<<etoile.bmp>>**		**<<formes.bmp>>**		**<<imori.bmp>>**	
	D_{KL}	D_{LL}	D_{KL}	D_{LL}	D_{KL}	D_{LL}	D_{KL}	D_{LL}
erreur de sous détection	0.0111	0.0063	0	0.0044	0	0.006	0.0001	0
erreur de sur détection	0.0726	0.0684	0.707	0.6568	0.6883	0.599	0.3755	0.374
mesure d'erreur de localisation	0.0349	0.0304	0.064	0.0597	0.054	0.059	0.007	0.012
Critère Lq	53.783	52.24	20.79	21.708	16.55	15.66	4.232	4.654
Critère de Bhattacharyya	0.072	0.566	0.785	0.2386	0.397	0.325	1.853	1.711
Critère de Kullback	118.373	110.4	49.21	48.92	38.59	34.83	9.960	10.88

Tableau V.3 : Résultats d'évaluation de la segmentation par PF D_{KL} et D_{LL}

- **Comparaison des resultats d'évaluation :**

critères	**<<woman.png>>**			**<<etoile.bmp>>**			**<<formes.bmp>>**			**<<imori.bmp>>**		
	canny	**Pf_ D_{LL}**	**P_ D_{KL}**	**canny**	**Pf_ D_{LL}**	**Pf_ D_{KL}**	**canny**	**Pf_ D_{LL}**	**Pf_ D_{KL}**	**canny**	**Pf_ D_{LL}**	**Pf_ D_{KL}**
erreur de sous détection		M				M			M		M	
erreur de sur détection	M			M				M		M		
mesure d'erreur de localisation	M	M		M	M		M	M		M	M	
Critère Lq	M			M				M				M
Critère de Bhattryya	M				M		M			M		
Critère de Kullback	M			M				M				M

Tableau V.4 : Comparaison des résultats de l'évaluation.
Avec (M : meilleure segmentation)

À partir de tableaux précédent, nous remarquons que la méthode de canny donne

des petits valeurs par rapport aux autres méthodes pour les différents critères utilisées. Donc, nous pouvons dire que la méthode de canny et plus performante que la méthode PF pour la segmentation des images en contours.

V .4 CONCLUSION :

Les différentes manipulations effectuées nous ont montré la difficulté de l'évaluation des méthodes de segmentation, Les deux algorithmes proposés ont donné des résultats différents pour les mêmes images. Sur la base des résultats obtenus, nous avons trouvé que la méthode de canny est plus performante que la méthode de filtrage paramétrique PF D_{KL} et D_{LL}.

Conclusion génerale

La segmentation d'images constitue une phase importante du processus d'analyse d'image. L'évaluation de la segmentation est indispensable pour classer les algorithmes développés et de connaître leur puissance. Certains opérateurs d'évaluation mesurent la qualité d'une segmentation, d'autres mesurent la différence entre l'image segmentée et une image de référence faites par des experts humains ou automatiquement par une autre méthode de segmentation.

L'objectif de notre travail est d'étudier quelques méthodes d'évaluation de la segmentation en contour d'images fixes. Pour ce but, nous avons utilisé deux méthodes de segmentation : la méthode de Canny el la méthode de filtrage paramétrique .Où les résultats d'évaluation de la segmentation effectuée montre que la méthode de canny présente de bonnes performances que la méthode du filtrage paramétrique.

Notre projet nous a été d'un grand bénéfice et intérêt. Il nous a permis d'enrichire nos connaissances dans le domaine de traitement d'image et la vision par ordinateur.

Bibliographie

[1] J. ANTOINE et L. JACQUES, Traitement de l'image, Université Catholique de Louvain; http://www.fyma.ucl.ac.be/files/UCL-IPT-04-24.pdf

[2] I. BLOCH , A. BOUCHER, F.TUPIN , Traitement et reconnaissance d'images, IFI. http://www.ensta-paristech.fr/~manzaner/Cours/IAD/AM_Introduction.pdf

[3] F. REJAUDRY, Comment interpréter l'histogramme d'une photo. http://www.pixelistes.com/download/Interpreter_un_histograme.pdf

[4] H. MAI TRE ,La détection des contours dans les images ; http://cours.etsmtl.ca/sys844/Documents/Lecture1.pdf

[5] M.HADAL-LAH ,Codage des images fixes par une méthode hybride basée sur la QV et les Approximations fractales, PFE USTHB, 1997.

[6] A.MANZANERA Traitement et reconnaissance d'images ,*Cours TERI – Master IAD UPMC Paris 6.* http://www.ensta-paristech.fr/~manzaner/Cours/IAD/AM_Filtrage.pdf

[7] E. ARNAUD, Extractions de caractéristiques - les contours. http://steep.inrialpes.fr/~**Arnaud**/vision_m2pgi/cours/c2_points_4pp.pdf

[8] H. MAITRE , La détection des contours dans les images. http://cours.etsmtl.ca/sys844/Documents/Lecture1.pdf

[9] J. MOSIG ,Traitement Détection de contours. http://morpheo.inrialpes.fr/people/Boyer/Teaching/**Mosig**/introduction.pdf

[10] A. SELAMI BOUALAM HAFIDHA Détection d'image fixes par la méthode de filtrage paramétrique et extension des méthodes à la décomposition en sous bande, UNIV-Medea2010.

[11] Y. AIT ALI YAHIA, A. GUESSOUM Détection de contour par la méthode de filtrage paramétrique, SETIT 2005.

[12] M.ALIKHOUDJA M.BENCHOHRA Segmentation des Images fixe à base d'Ondelettes, UNIV-MEDEA2011.

[13] S.AMIR Evaluation de la segmentation d'image sans référence, UNIV-BLIDA 2005.

[14] D. MINH CHOU Evaluation de la segmentation d'images, IFI 2007.

Annexes :

1-programme de canny :

```
clear all
clc
%inserer l'mage
load woman
%X=imread('image');
figure(1),clf,imshow(X);
%X=double(imread('cameraman.tif'));
%constant
N=4;
a0=1/4;
ech=2:-1:-2;
ech_l=length(ech);
SIZE=length(X);
% fonctions d'ondelettes
grad1=zeros(SIZE,SIZE);
Eg1=1;
for ec=1:ech_l
 i=ech(ec);
 a=1./a0^i
  for index_x=1:N;
     for index_y=1:N;
         x=index_x-(N+1)/2;
         y=index_y-(N+1)/2;
         xx=x.*a;
         yy=y.*a;
%Mexh
phi_x(index_x,index_y)=(1-2*xx.^ 2).*exp(-2*xx.^ 2)./sqrt(a);
phi_y(index_x,index_y)=(1-2*yy.^ 2).*exp(-2*yy.^ 2)./sqrt(a);
%Morl
% phi_x(index_x,index_y)=exp(-2.*(x.^2.*a)./2).*
cos(5.*x.*a)./sqrt(a);
% phi_y(index_x,index_y)=exp(-2.*(y.^2.*a)./2).*
cos(5.*y.*a)./sqrt(a) ;
%
%Gauss
% phi_x(index_x,index_y)=(1./2.*pi.*a.^2).*exp(-
(xx.*xx+yy.*yy)/(2.*a^2))
% phi_y(index_x,index_y)=(1./2.*pi.*a.^2).*exp(-
(xx.*xx+yy.*yy)/(2.*a^2))
%
end
   end;
```

```
phi_x=phi_x/norm(phi_x);
phi_y=phi_y/norm(phi_y);
% Faire la symétrie du noyau
phi_x=[phi_x(1:N/2,1:N);-phi_x(N/2+1:N,1:N)]
phi_y=[phi_y(1:N,1:N/2)';-phi_y(1:N,N/2+1:N)']'
% Affichage des deux noyau
figure(2);mesh(phi_x);title ('Noyau de ondelette suivant
x')
figure(3);mesh(phi_y);title ('Noyau de ondelette suivant
y')
% convolution
Gx1=conv2(X,phi_x,'same');
Gy1=conv2(X,phi_y,'same');
% module de gradient
Grads=sqrt((Gx1.*Gx1)+(Gy1.*Gy1));
%détermination des maximas locaux:
Eg=(mean(mean(Grads.^2)));
if Eg>Eg1
grad1=Grads;
Eg1=Eg;
echx(SIZE)=ech(ec);
end
   end
Grads=grad1;
% direction de gradient
angle_array=zeros(SIZE,SIZE);
for i=1:SIZE;
   for j=1:SIZE
      if (abs(Gx1(i,j))>eps*100)
      p=atan(Gy1(i,j)/Gx1(i,j))*180/pi;
        if (p<0)
        p=p+360;
end;
if (Gx1(i,j)<0 & p>180)
p=p-180;
  elseif (Gx1(i,j)<0 & p<180)
p=p+180;
   end
     else
     p=90;
       end
angle_array(i,j)=p;
        end
           end;
```

```
DD=zeros(SIZE,SIZE);
for i=2:SIZE-1
   for j=2:SIZE-1
      if ((DD(i,j)>=(-22.5) & angle_array(i,j)<=22.5) |
...
(angle_array(i,j)>=(180-22.5) &
angle_array(i,j)<=(180+22.5))) % 0/180
      if (Grads(i,j)>Grads(i+1,j) & Grads(i,j)>Grads(i-
1,j))
DD(i,j)=Grads(i,j);
      end
         elseif ((angle_array(i,j)>=(90-22.5) &
angle_array(i,j)<=(90+22.5)) | ...
(angle_array(i,j)>=(270-22.5) &
angle_array(i,j)<=(270+22.5))) % 90/270
         if (Grads(i,j)>Grads(i,j+1) &
Grads(i,j)>Grads(i,j-1))
DD(i,j)=Grads(i,j);
         end
          elseif ((angle_array(i,j)>=(45-22.5) &
angle_array(i,j)<=(45+22.5)) | ...
(angle_array(i,j)>=(225-22.5) &
angle_array(i,j)<=(225+22.5))) % 45/225
           if (Grads(i,j)>Grads(i+1,j+1) &
Grads(i,j)>Grads(i-1,j-1))
DD(i,j)=Grads(i,j);
               end
                else % 135/215
           if (Grads(i,j)>Grads(i+1,j-1) &
Grads(i,j)>Grads(i-1,j+1))
DD(i,j)=Grads(i,j);
     end
          end
               end
                   end
figure(4);imshow(DD);
```

2-programme de filtrage paramétrique DKL :

```
clear all;
format short e
% ******** lecture de l'Image originale:
XO =imread('image);
LS=4;N=LS;                    %la longueur du segment;
%
%*****calcul des paramètres ; alpha et neta:
%
theta=0.025*pi;               %la valeur de teta.
%
%*********spécification des neta:
%
na =-1;                       %la borne inferieur de
l'intervalle [na,nb]
nb =1;                        %la borne supérieure  de
l'intervalle [na,nb]
m=2;                          %le nombre d"éléments de
l"intervalle [na,nb]:=');
tic
for f=1:3
    X=XO(:,:,f);
    [l,c]=size(X);            % dimensions d'image
    L=l*c;                    % nombre total des éléments
de la matrice
    n=1;x(L)=0;
%
%*****transformation de ma matrice en une vecteur ligne:
%
    for ii=1:l
        for jj=1:c
            x(n)=X(ii,jj);
            n=n+1;
        end
    end
    x;
%
%****  Segmentation du vecteur Image et les aranger en
matrice segments suivant les lignes:
%
    for k=1:m;
        nk(k)=na+((k-1)*(nb-na))/(m-1);
    end
    Alpha=nk.*exp(j*theta);
    NS=L/LS;NS1=fix(NS);
    if NS > NS1
        NS=NS1+1;
        x(NS*LS)=0;
```

```
    end
    L1=NS*LS;
    ls=1;ns=1;xs(NS,LS)=0;
    for l1=1:L1
        xs(ns,ls)=x(l1);
        ls=ls+1;
        if ls>LS
            ls=1;
            ns=ns+1;
        end
    end
    xs;
%
%****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.
%
%******Traitement de tous les segments obtenus:
%
    for ns=1:NS
        xss(1:LS)=xs(ns,1:LS);
%
%****** les vecteurs convolutions obtenus dans les deux
cas:
%
        xss1=conv(xss,w1);%1/100*conv(xss,w1);
        xss2=conv(xss,w2);%1/100*conv(xss,w2);
%
%calcul de filtre suivant la relation yt=alpha1*yt-1+xt:
%
        for k=1:m;
            y0=20;
            y00=20;
            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(-j*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(-j*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:
%
    KKL=[];
    for i=1:NS
        KKL=[KKL KK1(i,1:LS)];
    end
    KKL=KKL';
%
%*****transformation du vecteur resultat en une matrice:
%
    ii=1;jj=1;XL(l,c)=0;
    for nn=1:L
```

```
        XL(ii,jj)=KKL(nn);
        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 une vevteur ligne:
%
    for ii=1:c
        for jj=1:l
            xc(n)=X(jj,ii);
            n=n+1;
        end
    end
    xc;
%
%****  Segmentation du vecteur Image et les aranger en
matrice segments suivant les lignes:
%
    ls=1;ns=1;xcs(NS,LS)=0;
    for l1=1:L1
        xcs(ns,ls)=xc(l1);
        ls=ls+1;
        if ls>LS
            ls=1;
            ns=ns+1;
        end
    end
    xcs;                % matrice segments
%
%******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);%1/100*conv(xcss,w1);
        xcss2=conv(xcss,w2);%1/100*conv(xcss,w2);
%
%calcul de filtre suivant la relation yt=alpha1*yt-1+xt:
%
        for k=1:m;
            y0=0;%20;
            y00=0;%20;
            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(-j*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(-j*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)=abs(Y1(k+1,ls)-Y1(k,ls));
                    P2(k,ls)=abs(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)=abs(1-Y1(k,ls));
                    P2(k,ls)=abs(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:
%
    KKC1=[];
    for i=1:NS
        KKC1=[KKC1 KK1(i,1:LS)];
    end
    KKC1=KKC1';
%
%******transformation du vecteur resultat en une matrice:
%
    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;
    XLC=(XL+XC)/2;%100*(XL+XC)/2;
    XLC1(:,:,f)=XLC;
end
% for  i=1:3
%     for j=1:l
%         for jj=1:c
%             q=XLC1(j,jj,i);
%             if q==0
%                 XLC2(j,jj,i)=256;
%             else
%                 XLC2(j,jj,i)=0;
%             end
%         end
%     end
% end
% XLC2=uint8(XLC2);
XLC1=uint8(XLC1);
Tex=toc
figure(1),clf,subplot(211),image(XO);subplot(212),image(X
LC1)
```

3-programme de filtrage paramétrique DLL :

```
clear;
format short e
% ******** lecture de l'Image originale:
XO =imread('image');
LS=4;                       %la longueur du segment;
theta=0.0003*pi;             %la valeur de la phase teta :
theta=-pi:0.001*pi:pi
na =-0.9;                   %borne inferieur de l'intervalle
nb =-na;                    %borne supérieure  de
l'intervalle
m=2;                        %nombre d"éléments del"intervalle
[na,nb]

N=LS;
tic
for f=1:3
        X=XO(:,:,f);
        [l,c]=size(X);                  % dimensions d'image
        L=l*c;                          % nombre total des
éléments de la matrice
        n=1;x(L)=0;
        %
        %transformation de ma matrice en une vevteur
ligne:
        %
        for ii=1:l
            for jj=1:c
                x(n)=X(ii,jj);
                n=n+1;
            end
        end
        x;
%
%****  Segmentation du vecteur Image et les aranger en
matrice segments suivant les lignes:
%

        for k=1:m;
            nk(k)=na+((k-1)*(nb-na))/(m-1);
        end
        Alpha=nk.*exp(j*theta);
        NS=L/LS;NS1=fix(NS);
        if NS > NS1
            NS=NS1+1;
            x(NS*LS)=0;
        end
        L1=NS*LS;
        ls=1;ns=1;xs(NS,LS)=0;
        for l1=1:L1
            xs(ns,ls)=x(l1);
            ls=ls+1;
```

```
            if ls>LS
                ls=1;
                ns=ns+1;
            end
        end
        xs;
%
%***détermination de la fenêttre de hamming symmétrique
et celle décalée:
%
        wh
=kaiser(LS);LS1=fix(LS/2);%hamming;bartlett;hann;blackman
        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.
%
%****Traitement de tous les segments obtenus:
%
        for ns=1:NS
            xss(1:LS)=xs(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(-
j*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(-
1i*theta)*A2(k,ls));
                    y0=y(k,ls);y00=y1(k,ls);
                end
            end
 %
 %******la méthode de D(l,l)suivant les lignes :
 %
            for ls=1:LS
                Sum1=0;
                for k=1:m
                    Y12(k,ls)=(abs(Y1(k,ls)-
Y2(k,ls))).^2;
                    Sum1=Sum1+Y12(k,ls);
                end
                YY=sqrt(N)*Sum1/m;
                KK1(ns,ls)=abs(YY);
            end
            KK1;
        end
  %
  %*****transformation de notre matrice de KK1 en
vecteur:
  %
        KKL=[];
        for i=1:NS
            KKL=[KKL KK1(i,1:LS)];
        end
  KKL=KKL';
%
%*****transformation du vecteur resultat en une matrice:
%
        ii=1;jj=1;XL(l,c)=0;
        for nn=1:L
            XL(ii,jj)=KKL(nn);
            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 une vevteur
ligne:
%
for ii=1:c
    for jj=1:l
        xc(n)=X(jj,ii);
        n=n+1;
    end
end
xc;
%
%****  Segmentation du vecteur Image et les aranger en
matrice segments suivant les lignes:
%
ls=1;ns=1;xcs(NS,LS)=0;
for l1=1:L1
    xcs(ns,ls)=xc(l1);
    ls=ls+1;
    if ls>LS
        ls=1;
        ns=ns+1;
    end
end
xcs;                % matrice segments
%
%******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(-j*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(-j*theta)*A2(k,ls));
            y0=y(k,ls);y00=y1(k,ls);
         end
    end
%
%*****la méthode de D(l,l)suivant les colonnes :
%
    for ls=1:LS
       Sum1=0;
       for k=1:m
           Y12(k,ls)=(abs(Y1(k,ls)-Y2(k,ls))).^2;
           Sum1=Sum1+Y12(k,ls);
       end
       YY=sqrt(N)*Sum1/m;
       KK1(ns,ls)=abs(YY);
   end
   KK1;
end
%
%*****transformation de notre matrice de KK1 en vecteur:
%
KKC1=[];
for i=1:NS
    KKC1=[KKC1 KK1(i,1:LS)];
end
KKC1=KKC1';
%
%******transformation du vecteur resultat en une matrice:
%
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;
XLC(:,:,f)=sqrt(abs(XL)+abs(XC));
end;
tt=toc
XLC1=10*uint8(XLC);
```

```
figure(2),clf,subplot(211),image(XO),subplot(212),image(X
LC1)
```

4 Les programmes d'évaluation :

a- Critère de mesure d'erreur de localisation

```
%im1 = Vérité terrain
%im2 = Image Segmente
%fond=valeur du fond
%contour=valeur du contour


im1=imread('Vérité terrain');
im2=imread('image segmenté');
fond=0;
contour=255;
[a,b]=find(im1==contour & im2==contour);
Nand=length(a);

[a,b]=find(im1==contour);
Ni=length(a);

[a,b]=find(im2==contour);
Na=length(a);

%texte=strcat('Nombre de pixels contour Image segmentée1:
',int2str(Ni));
%disp(texte);

%texte=strcat('Nombre de pixels contour Image segmentée2:
',int2str(Na));
%disp(texte);

%texte=strcat('Nombre de pixels communs:
',int2str(Nand));
%disp(texte);

%texte=strcat('Nombre de pixels sursegmentés:
',int2str(Na-Nand));
%disp(texte);

%texte=strcat('Nombre de pixels soussegmentés:
',int2str(Ni-Nand));
%disp(texte);
```

```
taille=size(im1);
L=taille(1);
H=taille(2);

[a,b]=find(im1==contour & im2==fond);
[c,d]=find(im1==fond & im2==contour);
res=(length(a)+length(c))/(L*H);
disp(strcat('Errloc=',num2str(res,10)));
```

b-programmer de Critère d'erreur de sur détection :

```
unction res=ErrSur(im1,im2,fond,contour)
%im1 = Vérité terrain
%im2 = Image Segmente
%fond=valeur du fond
%contour=valeur du contour

im1=imread('Vérité terrain');
im2=imread('image segmenté');
fond=0;
contour=255;
[a,b]=find(im1==contour & im2==contour);
Nand=length(a);

[a,b]=find(im1==contour);
Ni=length(a);

[a,b]=find(im2==contour);
Na=length(a);

%texte=strcat('Nombre de pixels contour Image segmentée1:
',int2str(Ni));
%disp(texte);

%texte=strcat('Nombre de pixels contour Image segmentée2:
',int2str(Na));
%disp(texte);

%texte=strcat('Nombre de pixels communs:
',int2str(Nand));
%disp(texte);

%texte=strcat('Nombre de pixels sursegmentés:
',int2str(Na-Nand));
%disp(texte);

%texte=strcat('Nombre de pixels soussegmentés:
',int2str(Ni-Nand));
```

```
%disp(texte);

taille=size(im1);
L=taille(1);
H=taille(2);

[a,b]=find(im1==fond & im2==contour);
[c,d]=find(im1==contour);
res=length(a)/(L*H-length(c));
disp(strcat('ErrSur=',num2str(res,10)));
```

c-Programme de critere Lq

```
im1,im2 : 2 images à comparer
%nx,ny   : largeur et hauteur des images
%q       : ordre

im1=imread('Vérité terrain');
im2=imread('image segmenté');
nx=256;
ny=256;
q=1;
contour=255;
fond=0;

[a,b]=find(im1==contour & im2==contour);
Nand=length(a);

[a,b]=find(im1==contour);
Ni=length(a);

[a,b]=find(im2==contour);
Na=length(a);

%texte=strcat('Nombre de pixels contour Image segmentée1:
',int2str(Ni));
%disp(texte);

%texte=strcat('Nombre de pixels contour Image segmentée2:
',int2str(Na));
%disp(texte);

%texte=strcat('Nombre de pixels communs:
',int2str(Nand));
%disp(texte);

%texte=strcat('Nombre de pixels sursegmentés:
',int2str(Na-Nand));
%disp(texte);
```

```
%texte=strcat('Nombre de pixels soussegmentés:
',int2str(Ni-Nand));
%disp(texte);

im1=double(im1);
im2=double(im2);
tmp=abs(im1-im2);
for i=2:q
   tmp=tmp.*tmp;
end;

somme=sum(sum(tmp));
taille=size(im1);
res=((1/(taille(1)*taille(2)))*somme)^(1/q);
disp(strcat('Lq',int2str(q),'=',num2str(res,10)));
```

d-programme de Critère de Kullback :

```
%function res=DKu(im1,im2)
%im1,im2 : 2 images à comparer
%nx,ny   : largeur et hauteur des images


im1=imread('Vérité terrain');
im2=imread('image segmenté');
nx=256;
ny=256;
contour=255;
fond=0;

[a,b]=find(im1==contour & im2==contour);
Nand=length(a);

[a,b]=find(im1==contour);
Ni=length(a);

[a,b]=find(im2==contour);
Na=length(a);

%texte=strcat('Nombre de pixels contour Image segmentée1:
',int2str(Ni));
%disp(texte);

%texte=strcat('Nombre de pixels contour Image segmentée2:
',int2str(Na));
%disp(texte);

%texte=strcat('Nombre de pixels communs:
',int2str(Nand));
%disp(texte);
```

```
%texte=strcat('Nombre de pixels sursegmentés:
',int2str(Na-Nand));
%disp(texte);

%texte=strcat('Nombre de pixels soussegmentés:
',int2str(Ni-Nand));
%disp(texte);

im1=double(im1)+1;
im2=double(im2)+1;
tmp=im1-im2;

tmp=tmp.*log10(im1./im2);

somme=sum(sum(tmp));
taille=size(im1);
res=(1/(taille(1)*taille(2)))*somme;
disp(strcat('DKu=',num2str(res,10)));
```

e-programme de Critère de Bhattacharyya :

```
%im1,im2 : 2 images à comparer
%nx,ny   : largeur et hauteur des images


im1=imread('Vérité terrain');
im2=imread('image segmenté');

nx=256;
ny=256;
contour=255;
fond=0;
%im1=255-double(im1);
%im2=255-double(im2);


[a,b]=find(im1==contour & im2==contour);
Nand=length(a);

[a,b]=find(im1==contour);
Ni=length(a);

[a,b]=find(im2==contour);
Na=length(a);

%texte=strcat('Nombre de pixels contour Image segmentée1:
',int2str(Ni));
%disp(texte);
```

```
%texte=strcat('Nombre de pixels contour Image segmentée2:
',int2str(Na));
%disp(texte);

%texte=strcat('Nombre de pixels communs:
',int2str(Nand));
%disp(texte);

%texte=strcat('Nombre de pixels sursegmentés:
',int2str(Na-Nand));
%disp(texte);

%texte=strcat('Nombre de pixels soussegmentés:
',int2str(Ni-Nand));
%disp(texte);

im1=double(im1);
im2=double(im2);
%tmp=im1-im2;

tmp=(im1.*im2).^(1/2);

somme=sum(sum(tmp));
taille=size(im1);
res=-log10((1/(taille(1)*taille(2)))*somme);
disp(strcat('DBh=',num2str(res,10)));
```