



D0008/04B

DEPARTEMENT D'ELECTRONIQUE

**THESE DE DOCTORAT
EN ELECTRONIQUE**

Présentée par :

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

LASSOUAOUI Nadia

Ingénieur d'Etat en Electronique
Magister en Electronique Option Signal et Communications

**Segmentation des Images par Différentes
Approches : Applications aux Images
Biomédicales**

Devant le Jury d'examen composé de:

Président :	BERKANI Daoud	Professeur	ENP	ALGER
Rapporteurs :	HAMAMI Latifa	Maître de conférences	ENP	ALGER
	ZERGUERRAS Ahmed	Professeur	ENP	ALGER
Examineurs :	GUESSOUM Abderrezak	Professeur	Université	BLIDA
	SMARA Youcef	Professeur	USTHB	ALGER
	BELOUHRANI Adel	Maître de Conférences	ENP	ALGER
	SERIR Amina	Maître de Conférences	USTHB	ALGER
Invités :	KHELLADI Abdelkader	Professeur		CERIST
	NOUALI Nadia	Chargée de Recherche		CERIST
	AÏT ABDELKADER Belaïd	Médecin Spécialiste en Biochimie		CPMC

DEDICACES



Je dédie le fruit de mes années de travail à la mémoire de ma très chère mère qui aurait été si fière de moi, ses conseils et mots m'ont toujours accompagné, que Dieu la garde dans son vaste paradis,

A mon très cher père, que dieu me le préserve,

A tous ceux qui me sont chers, et plus particulièrement à :

Mes sœurs, frères et leurs familles,

Madame Hamami, pour son amitié et sa confiance,

Ma sœur et amie Lamia et à sa petite famille,

Ma très chère nièce Nassima,

Mes amis et plus particulièrement Samira et son époux Abdel-Malek.

A tous ceux que j'aime, que Dieu le tout puissant me les garde.

Remerciements

Les résultats présentés dans cette thèse sont aussi l'aboutissement de recherche effectués au laboratoire des systèmes informatiques LSI, du Centre de Recherche sur l'Information Scientifique et Technique (CERIST) d'Alger.

J'exprime ma profonde gratitude et mes remerciements les plus sincères à mes directeurs de thèse Mme L. Hamami et M. A. Zerguerras, pour leurs soutiens, encouragements et précieux conseils.

J'exprime ma profonde gratitude à M. A. Khelladi, Directeur du CERIST, pour ses encouragements et pour avoir accepté d'être membre de mon jury.

Mes sincères remerciements à Mme Nadia Nouali, responsable du laboratoire LSI, pour son accueil, la confiance qu'elle m'a toujours témoignée et de participer au jury de cette thèse.

J'exprime ma profonde reconnaissance à M. D. Berkani, qui m'a fait l'honneur de présider le jury.

Je tiens aussi à exprimer mes remerciements et ma plus vive gratitude à M. Guessoum, M. Smara, M. Belouchrani, Mme Serir et M. Aït Abdelkader qui ont accepté de juger mon travail.

Mes sincères remerciements à M. Beghdadi, Professeur à l'université Paris 13, pour son accueil au sein de son laboratoire L2TI.

Mes remerciements chaleureux à tous mes collègues de : l'équipe LSI, M. Meftouh, la formation, la diffusion et la photocomposition du CERIST.

Que tous ceux qui ont contribué de près ou de loin à l'élaboration de cette thèse soient remerciés.

Résumé

Un des problèmes les plus importants en traitement et analyse d'images et obligatoire dans tous les systèmes de vision artificielle est la **segmentation** qui conditionne fortement la qualité de l'interprétation.

Cette thèse est née du fait qu'il n'existe pas actuellement de technique universelle de segmentation qui soit en mesure de traiter la multitude d'images réelles pouvant apparaître en pratique. Chaque technique est plus ou moins adaptée à un certain contexte. Un problème de choix et de paramétrage des opérateurs en fonction du contexte, naît alors.

En plus des techniques de segmentation classiques par 'contours' et en 'régions', nous avons exploré différentes théories, à savoir : logique floue, fractale, multifractale, morphologie mathématique,...

Les nombreux avantages qu'offrent les algorithmes génétiques, ont fait que nous nous sommes intéressés à leurs apports dans la segmentation des scènes 2 D. L'algorithme de relaxation sélectionniste a été étudié et appliqué sur des images de scènes réelles. Des améliorations ont été apportées par proposition d'un opérateur de mutation qui s'adapte aux propriétés spatiales de chaque pixel. Une combinaison de cet algorithme avec les multifractales a donné de meilleurs résultats. Nous avons utilisé cet algorithme hybride pour la segmentation des images de cellules biologiques du col utérin qui, notons-le, sont très complexes. La classification des cellules est aussi effectuée par utilisation des algorithmes d'étiquetage et un critère de mesure de la morphologie des cellules (RNC).

Mots clés : Segmentation d'images, détection de contours, régions homogènes, étiquetage, classification, images réelles, cellules biologiques du col.

Abstract

One of the most important problems in image processing and necessary in all artificial vision systems is the **segmentation** which influences strongly the quality of interpretation.

This thesis is defined from the fact that it does not exist an universal segmentation technique which is able to process the multitude of real images which can appear in practice. Each technique is more or less adapted to a certain context. Thus, the problem of finding the parameters of operators according to the context is stated. In addition to the traditional segmentation techniques by 'edges' and by 'regions', we explored various theories (e.g.: fuzzy logic, fractal, multifractal, mathematical morphology).

Considering the great advantages of the genetic algorithms, we were interested in their contributions in the segmentation of the scenes 2D. The selectionniste relaxation algorithm is studied and applied to real images. Improvements were made by proposing a mutation operator which adapts to the spacial properties of each pixel. We also combined this algorithm with the multifractals and better results are obtained. This hybrid algorithm is used for the segmentation of the biological cell images of the uterine collar, which are very complex. Also the classification of the cells is carried out with labelling algorithms and a criterion which measures their morphology RNC.

Key words : Image segmentation, edge detection, homogeneous areas, labelling, classification, real images, biological cells.

ملخص

تعد عملية التجزئة من أهم المشاكل في تحليل و معالجة الصور و من الخطوات الضرورية في جميع أنظمة الرؤية الاصطناعية التي تتوقف عليها نوعية ترجمة الصور. و كانت فكرة هذه المذكرة نتيجة لعدم وجود تقنية عالمية للتجزئة في الوقت الحالي التي بمقدورها معالجة معظم الصور الحقيقية التي تظهر في الواقع. فكل تقنية تلائم نوعا معينا من الصور، ومنه يطرح مشكل التقنية الملائمة و عواملها.

إضافة إلى التقنيات الكلاسيكية لتجزئة "الحدود" أو "المناطق" لجأنا إلى مختلف النظريات مثل: المنطق الغامض و فراكتال و متعدد فراكتال و المورفولوجية الرياضية و العديد من الالجابيات التي تقدمها الخوارزميات الوراثة، دفعتنا هذه الأخيرة إلى الاهتمام بما تقدمه في تجزئة المشاهد الثنائية الأبعاد (2D). فقمنا بدراسة و تطبيق خوارزمية التطابق الاصطناعي (relaxation sélectionniste) على صور لمشاهد حقيقية. وأدرجنا تحويلات من خلال تقديم معامل التحويل الذي تأقلم مع مختلف الخصائص الموضعية لكل النقاط. وأضفنا لهذه الخوارزمية تحليلا متعدد الفركتال وهو ما أدى إلى نتائج أفضل. فاستعملنا هذه الخوارزمية المهجنة في تجزئة الصور لخلايا بيولوجية ضمن عنق الرحم و التي تعتبر جد معقدة. كما قدمنا تصنيفاً لهذه الخلايا باستعمال خوارزمية التأشير و معيار لقياس مورفولوجيا الخلايا.

الكلمات الدالة : تجزئة الصور، تحديد الحدود، المناطق المتجانسة، التأشير، التصنيف، الصور الحقيقية، الخلايا

البيولوجية لعنق الرحم.

Table des Figures

Figure 3. Schéma d'un champ de mesure du variogramme	35
Figure 4. Détection de contours par la méthode vario-spectrale pour deux valeurs de θ .	36
Figure 5. L'approche multifractale pour la détection de contours.	39
Figure 6. Détection de contours par les gradients morphologiques, (a) image 2 dés, (b) gradient par érosion, (c) par dilatation et (d) symétrique, S : seuil.	42
Figure 7. Les éléments structurants L et E .	43
Figure 8. La ligne de partage des eaux.	45
Figure 9. Segmentation par seuillage des images (ai), avec (bi) l'algorithme automatique et (ci) l'algorithme de Fisher.	49
Figure 10. (a) Fonction S_z , (b) fonction d'appartenance non symétrique.	52
Figure 11. Fonctions limites	52
Figure 12. Segmentation par l'algorithme Fuzzy C-Means.	56
Figure 13. Segmentation par la relaxation probabiliste.	58
Figure 14. Segmentation de l'image 2 dés par l'algorithme de DH pour $T_\sigma=75$.	60
Figure 15. Segmentation des dés par l'algorithme de DH.	60
Figure 16. Segmentation de l'image muscle par l'algorithme de DH.	61
Figure 17. Segmentation de l'image cerveau par l'algorithme de DH.	61
Figure 18. Segmentation de l'image de cellules par l'algorithme de DH.	61
Tableau. Les intervalles obtenus par l'algorithme DH.	62
Figure 19. Segmentation multifractale de l'image 2 dés, $\alpha_{Min}=1.777$, $\alpha_{Max}=2.776$.	63
Figure 20. Segmentation multifractale de l'image des cellules biologiques du col, $\alpha_{Min}=1.928$, $\alpha_{Max}=2.729$.	64
Figure 21. Segmentation par l'opérateur morphologique géodésique.	65
Figure 22. Images résultantes de la segmentation en utilisant l'algorithme de reconstruction morphologique géodésique.	65
Figure 23. Equivalence entre régions.	68
Figure 24. Etiquetage des objets des images.	72
Figure 25. Organigramme de la méthode d'étiquetage par masquage récursif.	73

Chapitre 3 : Les Algorithmes Génétiques : Théorie et Applications en Vision

Figure 1. Roue de Loterie	80
Figure 2. Exemple de probabilités de sélection basée sur l'évaluation et le rang	81

pour différentes valeurs de N .	110
Figure 5. Segmentation par l'algorithme de relaxation sélectionniste.	114
Figure 6. Schéma de l'algorithme hybride de segmentation Multifractal-Génétique.	116
Figure 7. (ai) représente les images originales, (bi) les résultats de segmentation multifractale non supervisée et (ci) les résultats de segmentation multifractale-génétique.	117
Figure 8. Segmentation des images par la relaxation multifractale sélectionniste.	122
Tableau. Valeurs des seuils $S_{\alpha 1}$ et $S_{\alpha 2}$ des images de la figure 8.	123
Figure 9. (a) Image originale, (b) et (c) images des cytoplasmes et des noyaux, (d) et (e) cytoplasmes et noyaux étiquetés.	125
Figure 10. Classification des cellules du col.	127

Sommaire

Sommaire



Introduction Générale	2
Problématique	6
Chapitre 1: Analyse d'images : Etat de l'art sur la segmentation	
1.1 Introduction	8
1.2 Le processus d'analyse d'images	8
1.3 Critères d'évaluation	10
1.4 Approches frontières	13
1.4.1 Détection de contours par le calcul du gradient	13
1.4.2 Détection de contours par le Laplacien	17
1.4.3 Amincissement des lignes de contours	18
1.4.4 Seuillage	19
1.4.5 Le prolongement, la correction et le chaînage	19
1.5 Approches régions	21
1.5.1 Méthodes basées sur l'exploitation de l'histogramme	21
1.5.2 Croissances de régions	22
1.5.3 Segmentation par relaxation	24
1.6 Segmentation des images texturées	25
1.7 Coopération par fusion région/ contour	28
1.8 Conclusion	28
Chapitre 2: Segmentation des images : Algorithmes et applications	
2.1 Introduction	31
2.2 Approches de détection de contours	31
2.2.1 Détection de contours par la théorie fractale	31
2.2.1.1 Algorithme Spectral	32
2.2.1.2 Calcul du Variogramme	35
2.2.1.3 Méthode Vario-Spectrale	35
2.2.2 Détection de contours par la théorie multifractale	36
2.2.3 Détection de contours par la morphologie mathématique	39
2.2.3.1 Morphologie Mathématique à deux niveaux de gris	39
2.2.3.2 Morphologie mathématique à plusieurs niveaux de gris	41
2.2.3.3 Filtrage morphologique	41
2.2.3.4 Détection de contours par utilisation du gradient morphologique	42
2.2.3.5 L'amincissement morphologique	43
2.2.3.6 La ligne de partage des eaux	44
2.2.4 Autres approches de détection de contours	45
2.2.5 Conclusion	46
2.3 Approches de segmentation en régions homogènes	46
2.3.1 Techniques de seuillage	46
2.3.1.1 Maximisation de l'entropie a Posteriori	47
2.3.1.2 Algorithme de binarisation automatique	47
2.3.1.3 Algorithme multiseuillage de Fisher	48
2.3.2 Segmentation par la logique floue	50
2.3.2.1 Segmentation par seuillage floue	50
2.3.2.2 Algorithme Fuzzy C-Means	53
2.3.3 Algorithme de relaxation	56
2.3.4 Algorithme de dichotomie hiérarchique d'histogramme	59
2.3.5 Segmentation Multifractale	63

2.3.6	Segmentation par la morphologie mathématique	64
2.3.7	Conclusion	66
2.4	Segmentation par étiquetage en composantes connexes	67
2.4.1	Algorithme d'étiquetage par balayage séquentiel de l'image	68
2.4.1.1	La première phase	69
2.4.1.2	La deuxième phase	70
2.4.2	Algorithme de masquage récursif	71
2.5	Conclusion	74

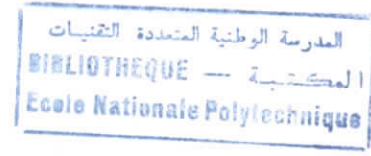
Chapitre 3: Les Algorithmes Génétiques : Théorie et Applications en Vision

3.1	Introduction	76
3.2	Fondements des algorithmes génétiques	77
3.2.1	La représentation chromosomique	77
3.2.2	La population initiale	78
3.2.3	La fonction d'évaluation	78
3.2.4	Les opérateurs génétiques	78
3.2.4.1	Sélection	79
3.2.4.2	Croisement	81
3.2.4.3	Mutation	83
3.2.5	La nouvelle population	84
3.3	Les paramètres d'un AG	84
3.3.1	La taille de la population	84
3.3.2	Le taux de croisement	84
3.3.3	Le taux de mutation	84
3.3.4	Le fossé des générations	85
3.3.5	Taux de diversité	85
3.4	Les étapes d'un AG	85
3.5	Les variantes du modèle génétique	86
3.5.1	L'algorithme génétique canonique	86
3.5.2	L'algorithme génétique modifié	87
3.6	Association des AGs avec des méthodes locales	88
3.7	Le parallélisme	88
3.7.1	Parallélisme par îlots	89
3.7.2	Parallélisation des calculs	89
3.8	Applications des algorithmes génétiques en vision	90
3.9	Conclusion	97

Chapitre 4: Applications des algorithmes génétiques en segmentation et classification des images cytologiques

4.1	Introduction	99
4.2	Segmentation avec un algorithme génétique : La relaxation sélectionniste	99
4.2.1	Description générale de la méthode	99
4.2.2	Les éléments de la méthode	100
4.2.3	Les étapes de l'algorithme	102
4.2.3.1	Initialisation	102
4.2.3.2	Sélection	102
4.2.3.3	Recombinaison et Mutation	103
4.2.4	Paramètres de l'algorithme	104
4.2.5	Variances locales de la mutation	106
4.2.6	Application à la segmentation des images de scènes réelles	109
4.3	Segmentations Multifractale-Génétique	115
4.3.1	Algorithme Multifractal-Génétique	115
4.3.2	Description de la méthode multifractale-génétique	116
4.3.2.1	Module de modélisation	117
4.3.2.2	Procédure de classification	118
4.3.3	Applications	118

4.4	Algorithme de relaxation multifractal-sélectionniste	119
4.5	Classification des cellules biologiques du col utérin	124
4.5.1	Calcul du rapport nucléo-cytoplasmique de chaque cellule	124
4.5.2	Classification selon le RNC	125
4.5.3	Application	125
4.6	Conclusion	128
	Conclusion Générale	130
	Bibliographie	133



Introduction Générale



Introduction Générale

Le cadre général dans lequel s'inscrit cette thèse est celui de la "segmentation d'images". La segmentation est un traitement bas niveau qui consiste à partitionner une image en régions homogènes (pour une ou plusieurs caractéristiques), chaque région ayant des caractéristiques différentes des régions voisines. La segmentation est un moyen perfectible de réduction de données, permettant d'extraire des primitives constituant des informations relatives au contenu de l'image, nécessaire pour un traitement de plus haut niveau. Cette opération constitue la base de tous les processus d'interprétation d'images et c'est à ce titre qu'elle reste encore l'un des principaux thèmes de recherche en vision par ordinateur.

Un grand nombre d'opérateurs ont été développés et peuvent être trouvés dans la littérature [1, 2, 3, 4]. L'application de différents algorithmes à une même image donne souvent des résultats différents. Cependant, une technique peut être plus adaptée à un type d'images ou à un certain problème donné qu'une autre. Aussi, la segmentation reste généralement dépendante : Des spécificités de l'image à traiter, des opérations situées en aval de la segmentation, des primitives à extraire, des contraintes d'exploitation, ...

Un problème supplémentaire est constitué par le fait que la plupart des méthodes existantes possèdent des paramètres qui nécessitent d'être réglés. Les valeurs attribuées aux paramètres d'un opérateur influencent de façon importante son comportement. En effet, de faibles variations peuvent impliquer des changements remarquables sur les résultats. Ainsi, la non-unicité, l'incertitude et l'instabilité des solutions par rapport aux données d'entrée font que la segmentation d'images reste un problème mal posé.

Du fait de cette diversité, il est difficile de définir de manière absolue une "bonne" segmentation. Ainsi, le problème de segmentation reste ouvert, surtout dans le cadre d'analyse d'images réelles riches en détails et contenant une grande variété de régions de nature uniforme ou texturée.

Comme il est difficile de trouver du premier coup la bonne collection de méthodes. Alors, en plus des techniques classiques de segmentation, nous avons étudié d'autres approches qui se basent sur des théories et principes différents. Nous avons ainsi exploité ces différentes théories (fractale [5], multifractale [6, 7], morphologie mathématique [8], logique floue, ...) et plus particulièrement les algorithmes génétiques [9, 10].

Les algorithmes génétiques sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisement, mutation, sélection, etc. Ils ont déjà une histoire relativement ancienne puisque les premiers travaux de John

Holland sur les systèmes adaptatifs remontent à 1962. L'ouvrage de David Goldberg [11] a largement contribué à les vulgariser.

Ils ont été appliqués à la segmentation d'images, soit, pour optimiser les paramètres d'un algorithme de segmentation existant, soit, pour rechercher une configuration optimale dans l'espace des segmentations possibles. Cependant, ils se heurtent à certaines difficultés, en particulier :

- ✓ Dans tous les cas, il est nécessaire de définir une fonction de fitness qui mesure la qualité de segmentation. Il n'existe pourtant pas de mesure communément admise, et des mesures distinctes peuvent conduire à des classifications distinctes. De plus, la fonction de fitness peut elle-même dépendre de paramètres qu'il faudra ajuster.
- ✓ Il est également nécessaire de déterminer les valeurs des paramètres qui régissent le comportement de ces algorithmes, cependant, l'état actuel de la théorie des AGs ne permettant pas de disposer de règles automatiques pour fixer les valeurs de ses paramètres, il est souvent nécessaire de procéder à un ajustement empirique.

Nous avons présenté et étudié en détail la relaxation sélectionniste qui dépasse les limitations déjà mentionnées [15, 16, 17]. Dans cette approche, la segmentation émerge progressivement des interactions qui se développent entre des unités porteuses de labels de régions et de vecteurs de caractéristiques en niveaux de gris. La sélection des unités sur la base de leur adéquation aux caractéristiques locales en chacun des points de l'image est l'élément clé de la méthode, d'où elle tire son nom de *relaxation sélectionniste*. L'intérêt de cette méthode est sa sensibilité aux petites variations, ainsi qu'à la détection des régions constituées de points dont l'intensité moyenne est très proche de celles des régions voisines.

Dans la version initiale avec laquelle elle a été proposée, l'opérateur de mutation procède par ajout d'une perturbation qui suit la loi normale. Pour simplifier l'algorithme tout en assurant une bonne qualité de segmentation et surtout une grande adaptabilité pour les différents types d'images, nous avons proposé l'utilisation d'un autre opérateur de mutation qui s'adapte aux propriétés spatiales de chaque pixel, aussi, pour assurer une convergence vers l'optimum global, la probabilité de mutation P_m suit une loi de décroissance en fonction de la génération.

Vu le grand nombre de types d'images de scènes réelles : médicales, satellites, ... Nous nous intéressons plus particulièrement aux images de cellules du col utérin, dont l'objectif de la segmentation est la séparation des noyaux et des cytoplasmes. La problématique du traitement de ce type d'images sera donnée.

Afin d'augmenter la qualité de segmentation par la relaxation sélectionniste et aussi de l'adapter à des applications bien précises, telle que la segmentation des cellules biologiques du col, nous avons proposé de lui associer l'algorithme multifractal. Une meilleure qualité de segmentation est obtenue.

Dans un souci de clarté, nous avons divisé le travail constituant notre thèse en quatre chapitres. Après la présentation de la problématique du traitement des images cytologiques, nous présentons ces différents chapitres :

- *Chapitre 1 : Analyse d'images : Etat de l'art sur la segmentation.* Ce chapitre s'intéresse à situer la segmentation par rapport aux différents traitements de l'analyse d'images. Un état de l'art sur les techniques de segmentation est donné.
- *Chapitre 2 : Segmentation des images : Algorithmes et Applications.* En plus des techniques classiques de détection de contours et de segmentation en régions homogènes, plusieurs autres techniques basées sur des principes différents sont présentées, tels que: l'utilisation de la dimension fractale, des exposants de Hölder, le gradient et la géodésie morphologiques, ... Des résultats d'application sur des images de scènes réelles sont donnés. Des améliorations sont proposées soit pour remédier à certains inconvénients, soit pour simplifier l'utilisation des algorithmes. La segmentation par classification ou étiquetage des régions connexes a été aussi présentée. Des résultats d'application sur différents types d'images sont donnés.
- *Chapitre 3 : Les algorithmes génétiques : Théorie et applications en vision.* Ce chapitre nous permettra de nous familiariser avec les algorithmes génétiques. Ainsi, la théorie, les fondements et les opérateurs sont donnés. Aussi, Les travaux effectués par les algorithmes génétiques dans la vision sont présentés.
- *Chapitre 4 : Applications des algorithmes génétiques en segmentation et classification des images cytologiques.* L'algorithme de relaxation sélectionniste basé sur les opérateurs génétiques est étudié. Deux algorithmes hybrides Multifractal-génétique pour la segmentation et la classification des images de cellules du col utérin sont proposés.

Après présentation et discussions de tous les aspects de la segmentation dans les chapitres cités, nous terminerons, par une conclusion générale.

Problématique

Problématique

L'application cytologique existe en vision par ordinateur depuis les années soixante [18, 19]. Malgré les travaux qui se sont succédés, les difficultés posées par celle-ci n'ont toujours pas été résolues. Les difficultés et la complexité des images de cellules biologiques sont dues à ces principales caractéristiques :

- Le contraste entre les cellules et le fond de l'image est très faible.
- L'intensité du fond de l'image est non uniforme.
- A l'intérieur même d'une même cellule, des variations importantes d'intensité existent.
- La frontière exacte de la cellule est floue (les bords de la cellule).

Pour reconnaître des cellules, il est apparu judicieux de les diviser en différentes composantes, puis les utiliser pour la reconnaissance de chaque cellule. Les trois principales composantes cellulaires sont :

1. Le noyau qui est le cœur de la cellule.
2. Le cytoplasme qui représente toute la partie de la cellule qui entoure le noyau. Sa segmentation est très difficile, car il est très peu homogène et le contraste entre celui-ci et le fond est très faible.
3. Le fond de l'image entoure les cellules, il est bruité et non uniforme.

Ainsi, pour la reconnaissance, il est nécessaire de détecter les noyaux et les cytoplasmes séparément. Cependant, la segmentation des cellules biologiques [18, 19, 20] est l'un des exemples d'application de vision où les techniques de segmentation ne donnent pas facilement une bonne qualité. Puisque les images de cellules présentent beaucoup d'hétérogénéité surtout au niveau du cytoplasme et du fond. Aussi, les frontières des cytoplasmes sont souvent floues, imparfaites, interrompues et supportées par de faibles contrastes, mal définies puisque les cellules baignent dans le fond qui est extrêmement hétérogène. D'autre part, il est rare que les cellules soient isolées : elles forment le plus souvent des amas.

Il existe aussi quatre types de cellules dans une coupe muqueuse vaginale qui sont : Les basales, para-basales, intermédiaires et superficielles [21]. L'analyse morphologique des cellules du col montre que la taille des noyaux et des cytoplasmes diffèrent d'une couche à une autre [21]. Et le rapport nucléo-cytoplasmique RNC permet de mesurer cette différence morphologique entre les cellules des différentes couches.

Ainsi, pour des besoins de reconnaissances des anomalies, il est nécessaire de procéder par :

1. Après l'acquisition des images de cellules récoltées sur lamelle grâce au frottis cervical en utilisant un microscope doté d'un appareil numérique, il est nécessaire de **prétraiter** les images.

2. Après le prétraitement, la séparation de chaque cellule en ses deux composantes principales noyau et cytoplasme doit être effectuée par la **segmentation**.
3. Ensuite, une étape de **classification** selon le type des cellules est nécessaire pour organiser la recherche dans l'étape de reconnaissance.
4. A partir de la segmentation et la classification, nous pouvons réaliser la **reconnaissance** pour la prise de décision à propos de la présence de l'anomalie.

Dans le cadre de notre travail, nous sommes intéressés par la conception des modules de prétraitement, segmentation et classification.

Pour le prétraitement, il faudra procéder par des traitements supervisés. Nous pouvons citer ces exemples :

- Il faudra rehausser la texture des cytoplasmes pour que les limites des cellules apparaissent mieux.
- Filtrer des formes bien spécifiques. Par exemple, pour éliminer des filaments, il faudra les rechercher sur l'ensemble de l'image grâce à un masque qui code la structure morphologique d'un filament. Une fois qu'il est détecté, il est confondu en lui affectant l'intensité moyenne de son voisinage.
- Procéder par la séparation des cellules qui se chevauchent ou qui se touchent.
- Eliminer les cellules tronquées.
- ...

A cause du faible contraste entre le fond et les cytoplasmes des cellules, il faudra choisir un algorithme de segmentation qui soit sensible même aux petites variations d'intensité moyenne des régions, pour une meilleure détection des cytoplasmes. Surtout que le respect des limites des corps cellulaires est important pour le diagnostic.

Pour la classification, nous exploitons les images des noyaux et cytoplasmes obtenues par la segmentation¹, un algorithme d'étiquetage doit être appliqué pour séparer chaque corps cellulaire (noyau ou cytoplasme) d'un autre dans l'image des noyaux ou cytoplasmes. L'objectif est d'estimer la surface de chacun pour obtenir le RNC de chaque cellule et ainsi son type : para-basal, basal, intermédiaire ou superficiel.

Dans les chapitres suivants, nous présentons en détails, les algorithmes choisis pour le traitement des images cytologiques.

¹ Des prétraitements peuvent être appliqués après la segmentation. Tel que : Le remplissage des trous ou l'élimination des petites régions aléatoires qui peuvent apparaître.

Chapitre 1

Analyse d'Images : Etat de l'Art sur la Segmentation

- 1.1 Introduction
- 1.2 Le processus d'analyse d'images
 - 1.3 Critères d'évaluation
 - 1.4 Approches frontières
 - 1.5 Approches régions
- 1.6 Segmentation des images texturées
- 1.7 Coopération par fusion région/ contour
 - 1.8 Conclusion

Analyse d'Images : Etat de l'Art sur la Segmentation

1.1 Introduction

Isoler les différentes entités qui composent une image, constitue un des problèmes les plus importants en traitement et analyse d'images. Cette opération, quasi obligatoire dans tous les systèmes de vision artificielle est la **segmentation**. Celle-ci conditionne fortement la qualité de l'interprétation, ce qui justifie les nombreux travaux qui lui ont été consacrés. Les premiers travaux remontent aux années soixante et se poursuivent encore de nos jours [1, 2, 3, 4]. Les espoirs étaient grands de réaliser un système de vision performant en s'aidant des techniques de segmentation. Cependant après une période d'évolution rapide, les progrès se sont ralentis et les « vieilles » techniques ont bien peu à envier aux méthodes actuelles. Après tant d'années de travail, les chercheurs butent toujours sur le problème de la segmentation.

Un grand nombre d'opérateurs de segmentation ont été développés et peuvent être trouvés dans la littérature [1, 2, 3, 4]. Malgré la complexité de certains de ces opérateurs, il n'y a aucune technique qui se détache pour présenter le titre de "meilleure méthode". Chacune d'entre elles possède des avantages et des inconvénients. Une technique peut être plus adaptée à un type d'images ou à un certain problème donné qu'une autre. Il se pose alors un problème de choix de l'opérateur à appliquer en fonction d'une variété d'informations. Et en général, on ne dispose que des informations de niveau de gris ou de couleur pour opérer la segmentation. Il existe deux principales approches en segmentation : l'approche frontière et l'approche région [3, 22].

Dans ce présent chapitre, nous présentons d'abord un schéma du processus d'analyse d'images pour situer la segmentation par rapport aux autres traitements, ensuite nous présenterons des définitions de bases sur la segmentation, aussi un état de l'art sur les méthodes classiques de segmentation sera donné.

1.2 Le processus d'analyse d'images

Le processus d'analyse d'images a pour but de fournir une **description** ou une **interprétation** d'une scène à partir de l'information extraite de l'image [3, 22, 23].

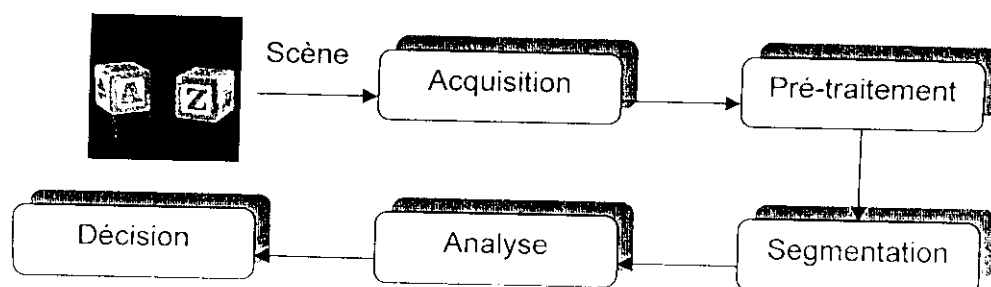


Figure 1 : Etapes d'un processus d'analyse d'images.

Acquisition : Au début, nous devons faire l'acquisition d'une scène, en discrétisant l'image réelle continue. Normalement, la quantité d'information brute initiale, après la discrétisation, est très volumineuse et difficile à manipuler. De plus cette discrétisation entraîne une perte d'information, de même qu'elle pose des problèmes d'ordre technique, comme l'illumination et la texture, entre autres.

Dans la suite de notre travail, la notion d'image utilisée est de nature bidimensionnelle discrète en plusieurs niveaux de gris. Donc, à chaque élément de l'image, c.-à-d., pixel, correspond un niveau d'intensité lumineuse appelé niveau de gris appartenant à $\{0, 1, \dots, N-1\}$ où 0 correspond à une illumination nulle (la couleur noire), $N-1$ correspond à l'intensité lumineuse maximale (la couleur blanche) et les autres valeurs sont des nuances de gris entre le noir et le blanc.

Des images brutes mémorisées par un système de vision contiennent une quantité énorme d'information. Cette quantité souvent trop importante peut être réduite en utilisant des procédures de pré-traitements.

Prétraitement : Etant *bruitées* ou *floues*, les images doivent passer par un *pré-traitement* avant toute opération de détection, où l'information dégradée est restaurée. Dans cette étape, des détails de l'image peuvent être rehaussés par plusieurs techniques [3].

Le pré-traitement consiste à sélectionner l'information nécessaire à l'application. Cette sélection passe souvent par l'élimination du bruit dû aux conditions d'acquisition, par la normalisation des données ainsi que par l'homogénéisation, par la suppression des informations redondantes, superflues et inutiles pour l'application visée [18].

Les opérations de pré-traitement peuvent être classées en trois groupes [3, 24] :

1. Les opérations ponctuelles qui modifient ponctuellement les valeurs des niveaux de gris des pixels.
2. Les opérations sur voisinages où la nouvelle valeur du pixel est obtenue à partir des valeurs des niveaux de gris de ses voisins.
3. Les opérations globales où l'étendue de la transformation est toute l'image pixel par pixel.

Le filtrage est une opération de pré-traitement qui, à un signal d'entrée x appartenant à un espace ξ , associe un signal y appartenant à un sous-espace ζ de ξ . Souvent, il est utilisé pour l'extraction de l'information d'un signal bruité.

Après l'élimination des informations superflues et inutiles par le pré-traitement, l'image peut être segmentée.

La segmentation d'images : Elle est la phase la plus importante et la plus difficile du processus d'analyse d'images. Segmenter une image correspond à trouver les régions qui ont un sens, c.-à-d., extraire les objets constituant la scène, et plus précisément, le support de l'image devra être partitionné en éléments de surface, de manière à ce que chaque élément

correspondre à une entité au niveau de la scène analysée. A chaque entité localisée est associée une composante connexe étiquetée. Cette étape doit permettre d'interpréter la scène aussi bien que le ferait l'observateur ; A partir des résultats fournis par la segmentation [3, 22], viennent les **traitements de haut niveau**, tels que la description de l'image, la reconnaissance des formes et les prises de décisions.

Zucker [2] a énoncé la définition de la segmentation comme suit :

Etant donné un prédicat¹ P pouvant s'appliquer aux attributs des régions, une segmentation est un partitionnement de l'image R en régions R_i , telles que :

1. $R_i \neq \emptyset \forall i$
2. $R_i \cap R_j = \emptyset$
3. $R = \bigcup_i R_i$
4. R_i est connexe $\forall i$
5. $P(R_i) = \text{vrai}$ quel que soit i .
6. $P(R_i \cup R_j) = \text{faux}$ si $i \neq j$ et R_i et R_j sont adjacentes.

Cette définition n'est cependant pas totalement satisfaisante puisque la solution n'est pas nécessairement unique [23]. Deux segmentations ne différant que par quelques pixels peuvent raisonnablement être considérées comme équivalentes, mais des solutions présentant des différences beaucoup plus importantes, à l'échelle des régions, peuvent également satisfaire cette définition. Une même image peut en effet conduire à des segmentations très différentes suivant l'ordre dans lequel ses régions seront traitées.

La formalisation du problème de segmentation présente donc une réelle difficulté. Celle-ci peut être contournée par l'adoption d'un critère d'évaluation.

1.3 Critères d'évaluation

Face à la difficulté que représente la définition de la segmentation, une définition implicite peut être adoptée à travers le choix d'une mesure de qualité des images segmentées. En effet, une fois un tel critère défini, l'objectif de la segmentation est de rechercher l'image segmentée qui optimise ce critère.

Il existe de nombreuses mesures d'évaluation des techniques de segmentation [17, 25, 26, 27, 28, 29]. Il existe deux catégories de méthodes d'évaluation des algorithmes de segmentation : Les méthodes analytiques et les méthodes empiriques.

Les méthodes analytiques examinent directement les algorithmes de segmentation en analysant leurs principes et propriétés. Cependant, les méthodes empiriques jugent indirectement les algorithmes de segmentation en les appliquant à des images test et en mesurant la qualité des résultats de segmentation.

¹ Le prédicat est un critère d'homogénéité qui peut être fondé par exemple sur la luminance, ou l'indice de texture

- ✓ Notons qu'il n'est pas possible d'avoir toutes les propriétés des algorithmes de segmentation par des études analytiques, et jusqu'à maintenant, les méthodes analytiques traitent uniquement certains modèles particuliers ou des algorithmes de propriétés désirables.

Les méthodes analytiques peuvent dans certains cas donner une information quantitative à propos des algorithmes de segmentation. L'analyse des performances de plusieurs détecteurs de contours en utilisant le rapport de probabilité de détection dans une procédure de conception statistique existe. Soit T le seuil de décision du contour, P_c la probabilité de la bonne détection et P_f la probabilité de fausse détection :

$$P_c = \int_T^{\infty} p(t/\text{contour}) dt \quad (1)$$

$$P_f = \int_T^{\infty} p(t/\text{non-contour}) dt$$

La courbe $P_c = P_f(T)$ fournit l'index de performance des détecteurs.

D'autres propriétés des algorithmes de segmentation peuvent être obtenues par analyse qui inclue la stratégie, la complexité et l'efficacité du traitement, ainsi que la résolution de l'algorithme de segmentation. Ces propriétés peuvent aider la sélection des algorithmes adéquats pour des applications particulières. Par exemple, la stratégie du traitement des algorithmes de segmentation peut être parallèle, séquentielle, itérative ; et les algorithmes parallèles sont adéquats pour des implémentation rapides.

Outre les méthodes analytiques, il existe des méthodes d'évaluation avec ou sans segmentation de référence ainsi que des méthodes fondées sur l'objectif pour lequel la segmentation a été entreprise [17, 25].

- ✓ L'une des mesures les plus simples de l'erreur réalisée par une segmentation est le pourcentage de pixels mal classés. Elle nécessite donc le recours à une segmentation de référence, dont on dispose généralement lorsqu'on utilise des images-test artificielles ou lorsqu'une segmentation manuelle peut facilement être réalisée. L'un des avantages de ce critère est la facilité avec laquelle il peut être calculé.

Pour pondérer les erreurs en fonction de leurs positions, un critère d'évaluation est utilisé comme la somme

$$\varepsilon = \sum_m d_m^2 \quad (2)$$

où d_m est la distance entre chaque pixel erroné m et le point le plus proche de même label dans la segmentation cible.

Avec un tel critère, on considère donc qu'attribuer un mauvais label à un pixel est une erreur d'autant plus importante que ce pixel se trouve loin de la région à laquelle ce label correspond réellement.

- ✓ Dans certaines situations, on ne dispose pas d'une segmentation de référence et les mesures d'erreurs définies précédemment sont inutilisables. Il est alors nécessaire de disposer de critères d'évaluation ne reposant pas sur une comparaison avec une segmentation de référence.

Levine et Nazif [28] définissent deux mesures de qualité d'une segmentation. La première est l'uniformité des régions. Elle est d'autant meilleure que l'hétérogénéité, définie comme une somme pondérée des variances des attributs sur les régions, est faible. La seconde est le contraste entre régions, fondé sur les différences des valeurs moyennes des attributs entre régions adjacentes.

Pour une image en niveaux de gris $f(x,y)$, soit R_i la i^{eme} région segmentée, A_i la zone de R_i , alors la mesure d'uniformité en niveaux de gris GU de $f(x,y)$ est :

$$GU = \sum_i \sum_{(x,y) \in R_i} \left[f(x,y) - \frac{1}{A_i} \sum_{(x,y) \in R_i} f(x,y) \right]^2 \quad (3)$$

Une normalisation de cette mesure a été proposée par Sahoo [25] et est donnée par :

$$1 - GU/C \quad (4)$$

C étant un facteur de normalisation.

Soit f_o la moyenne des niveaux de gris de l'objet, et f_b la moyenne des niveaux de gris du fond, la mesure du contraste peut être donnée par :

$$\frac{|f_o - f_b|}{f_o + f_b} \quad (5)$$

- ✓ On peut évaluer une segmentation, non par elle-même, mais en fonction de l'objectif pour lequel elle a été réalisée. Donc, une segmentation est plus ou moins bonne suivant qu'elle permette ou non de se rapprocher du but recherché. Le coût d'une segmentation est défini en fonction du nombre d'opérations de fusion et de division de régions nécessaires pour obtenir une segmentation correcte à partir de la segmentation considérée. Une segmentation est d'autant meilleure que le nombre de modifications qu'elle doit subir est faible.

D'autres critères d'évaluation existent [25, 26, 27, 28, 29], tels que : le nombre d'objets dans l'image, l'analyse de la forme de la région, ...

Les méthodes de segmentation existantes peuvent être séparées en deux groupes : segmentation par régions (où les pixels de chaque région sont connectés par l'uniformité de leurs caractéristiques, telles que : luminance, couleur, coordonnées, ...) et segmentation par contours (basée sur la recherche des discontinuités locales).

Dans les sections suivantes, nous présentons en détails ces deux grandes classes de méthodes de segmentation.

1.4 Approches frontières

Ces techniques ne conduisent pas à des segmentations au sens de la définition en termes de partition en régions homogènes. Elles fournissent en effet des images binaires dans lesquelles les points de contours sont distingués des autres sites.

La détection de contours peut être perçue différemment selon le but recherché. Ainsi, selon Pavlidis [23, 30], elle consiste en une détection des discontinuités présentes dans une image. La détection des contours est alors considérée comme l'une des premières étapes de la vision par ordinateur et aucune information *a priori* sur les objets présents dans l'image n'est utilisée. Toutefois, dans certaines applications, il est utile d'exploiter des informations sur la forme et l'emplacement approximatif de l'objet pour en détecter le contour.

Pour la détection des contours préparant la reconnaissance, il ne s'agit pas en fait de détecter toutes les transitions de niveaux de gris, mais plutôt celles qui permettent une description pertinente de l'image. Ainsi, un des problèmes fondamentaux s'avère être la définition même des discontinuités recherchées et la modélisation qui en résulte.

Un contour est une frontière entre deux différents milieux (deux couleurs, deux niveaux de gris, ...) ou une rupture (une discontinuité) locale d'une propriété intéressante d'un groupe de pixels. Pour déterminer la position des éventuels contours, on utilise des méthodes dérivatives. En effet, l'image dérivée met en évidence les variations de niveaux de gris. Quand il y a une forte variation, cela est souvent dû à un changement de teinte, d'éclaircissement ou de texture et de façon générale, à un changement de zone.

Les premières idées de modélisation d'une discontinuité étaient très simples. Une discontinuité était assimilée à un fort gradient ou à un passage par zéro du laplacien.

Nous montrons ici les deux approches, puis nous présentons d'autres détecteurs pour donner une vue d'ensemble des travaux qui ont été effectués dans ce domaine.

1.4.1 Détection de contours par le calcul du gradient

Dans le cas bidimensionnel, l'image est représentée par une fonction scalaire continue $A(x,y)$. Le vecteur gradient est défini au point M de coordonnées (x,y) par :

$$\nabla A(x,y) = \left(\frac{\partial A}{\partial x} \quad \frac{\partial A}{\partial y} \right)^t \quad (6)$$

Le vecteur gradient en un point M est normal à la courbe de niveau donné par $A(x,y)=\text{Constante}$ qui passe par M . La plus grande variation de $A(x,y)$ a lieu quand on se déplace le long de la normale à la courbe de niveau. Le maximum de cette variation est défini en grandeur et en direction par $|\nabla A(x,y)|$.

Au point $M(x,y)$, l'orientation ϕ du gradient est donnée par :

$$\phi = \arctan \left(\frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}} \right) \quad (7)$$

et le vecteur unitaire \mathbf{n} normal au contour et l'orientation du contour ont pour expression :

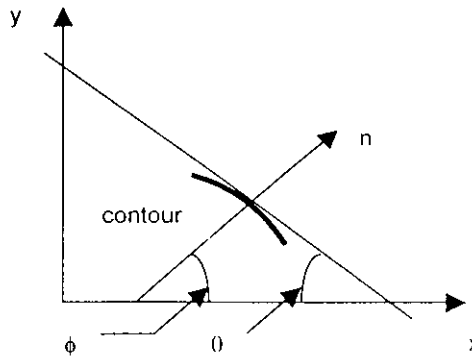


Figure 2 : Orientation d'un contour.

Il existe un nombre considérable de techniques pour calculer le gradient discret. Dans la plupart des cas, le gradient est calculé grâce à un produit de convolution d'un filtre h et de l'image A . Les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$A_j[i, j] = h_j * A[i, j] \quad \text{ct} \quad A_i[i, j] = h_i * A[i, j] \quad (9)$$

* : désigne l'opération de convolution,

La norme du gradient est donnée par :

$$|\nabla A[i, j]| = \sqrt{A_j^2[i, j] + A_i^2[i, j]} \quad (10)$$

ou encore :

$$|\nabla A[i, j]| = \max\{|A_j[i, j]|, |A_i[i, j]|\} \quad (11)$$

ou encore :

$$|\nabla A[i, j]| = \frac{\{|A_j[i, j]| + |A_i[i, j]|\}}{2} \quad (12)$$

Notons que les formules (11) et (12) sont plus intéressantes que la formule (10), puisqu'elles nécessitent moins d'opérations de calcul donc un temps d'exécution plus faible.

Plusieurs masques peuvent être utilisés dont :

Roberts :

$$h_i = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad h_j = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad (13)$$

Prewitt (c=1) et Sobel (c=2) :

$$h_i = \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix} \quad h_j = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \quad (14)$$

approximatives du gradient. Nous donnons les masques 3x3 et 5x5 de Kirsh.

Masques 3x3 de Kirsh :

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \text{ et } \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} \quad (15)$$

Masques 5x5 de Kirsh :

$$\begin{pmatrix} -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & -1 & 0 \end{pmatrix} \text{ et } \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 \\ 0 & -1 & -1 & -1 & -1 \end{pmatrix} \quad (16)$$

Opérateur MDIF :

Il s'agit de la combinaison d'un filtre moyenneur du noyau m et d'un dérivateur utilisant les masques directionnels de Prewitt h_i et h_j [22] :

$$m = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, h = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, h_i = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad (17)$$

Les approximations des composantes du gradient peuvent être calculées après le lissage ou directement en appliquant les masques suivants de taille 5x5 :

$$m_i = m * h_i = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{pmatrix}, m_j = m * h_j = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix} \quad (18)$$

Opérateur NAGDIF :

L'opérateur NAGDIF est la combinaison d'un lissage non linéaire de Nagao suivi d'une opération de dérivation [22].

Le filtrage de Nagao est un filtrage par fractionnement de la fenêtre d'analyse, la sélection des pixels intervenant dans le calcul de la sortie du filtre est faite en prenant en compte des contraintes géométriques.

L'opérateur de Nagao travaille sur un voisinage 5x5 centré sur le pixel courant. Dans ce voisinage, 9 domaines sont définis. Pour chaque domaine D_i , la moyenne μ_i et la variance $var(D_i)$ sont calculées. L'intensité du pixel courant est remplacée par la moyenne du domaine qui présente la plus faible variance.

			1	1
		1	1	1
			1	1

Domaines D_1 , D_2 et D_4 sont obtenus par rotation de $\pi/2$.

			1	1
		1	1	1
		1	1	1

Domaines D_5 , D_6 et D_8 sont obtenus par rotation de $\pi/2$.

	1	1	1	
	1	1	1	
	1	1	1	

Domaine D_9

Figure 3: Domaines de Nagao.

Cet opérateur étend une technique fondée sur un lissage non-linéaire avec des domaines carrés. Il présente l'avantage de réduire le bruit tout en renforçant les contrastes par diminution de la largeur de la zone de transition. Le domaine D_9 peut être ignoré sans altérer notablement les performances.

La dérivation pour l'opérateur du NAGDIF est effectuée par l'opérateur suivant :

$$CONTOUR \text{ Si } \max\{|B[i, j] - B[i - \varepsilon_1, j - \varepsilon_2]|\} > Seuil \quad (19)$$

Où ε_1 et ε_2 sont des valeurs entières de l'intervalle $[-1, 1]$ et B étant l'image lissée par le filtre de Nagao.

1.4.2 Détection de contours par le Laplacien

Les contours peuvent être aussi localisés par les passages par zéro du laplacien ou de la dérivée seconde dans une direction donnée:

$$\Delta A(x,y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} \quad (20)$$

En effet, tous les extremums du gradient correspondent à des zéros dans la dérivée de celui-ci. Pour ne pas être sensible aux valeurs minimales du gradient, c.-à-d., les valeurs nulles correspondant à des zones homogènes, il est nécessaire de détecter les passages par zéro du laplacien et non pas seulement les valeurs nulles de celui-ci.

Des approximations discrètes du laplacien, calculées sur un voisinage 3x3, correspondent aux masques suivants :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \text{ou} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (21)$$

Le problème majeur des techniques exploitant le laplacien est la sensibilité aux petites variations non significatives dues essentiellement au bruit. En effet, un des problèmes fondamentaux de la détection des contours est la résistance au bruit et aux petites variations locales dues à la texture ou à des irrégularités de surface. Pour remédier à ce problème, il est possible d'effectuer un prétraitement sur l'image afin de lisser les petites irrégularités locales. Ce prétraitement peut être réalisé par des filtres de différentes natures [3, 22, 31].

Pour plus de sûreté, il est aussi possible d'éviter les passages par zéro dans les zones avec variations locales faibles et irrégulières en acceptant uniquement les pixels dont la valeur du gradient est supérieure à un seuil [4].

Les opérateurs du premier et du second ordre sont intéressants et donnent des résultats satisfaisants dans le cas d'images simples, ce qui n'est pas vrai pour les images complexes.

Autres opérateurs du deuxième ordre

Marr et Hildreth [22, 23] suggèrent par similitude avec l'œil humain, l'utilisation d'un opérateur Laplacien-Gaussien. La méthode proposée consiste à convoluer l'image par une gaussienne avant d'appliquer un filtre Laplacien. Ils proposent l'utilisation de plusieurs masques correspondant à différentes valeurs de la variance σ^2 de la gaussienne. Aussi, Huertas et Médioni [22] proposent un filtre qui est une décomposition en filtres séparables de l'opérateur laplacien d'une gaussienne.

Cependant, les masques de variance trop large dégradent l'image et ne permettent pas la détection des structures fines, à l'inverse, une variance trop faible conduit à une détection du bruit trop importante.

Pour surmonter des difficultés de précision, de localisation et d'efficacité de détection, Canny [32] propose de formaliser mathématiquement une approche optimale de détection. Pour

déterminer le filtre optimal, il définit trois critères : bonne détection, bonne localisation et non multiplicité des réponses.

Trois filtres optimaux majeurs ont été obtenus en prenant en compte ces trois critères : le filtre de Canny, le filtre de Deriche et le filtre de Shen et Castan [22].

Canny trouve la solution suivante, qui est la dérivation de la convolution d'une image avec une gaussienne. L'approche proposée sert toujours de référence dans la littérature car sa solution est correctement formalisée et le comportement du filtre est ainsi connu avec précision.

Deriche, utilisant la même démarche que Canny a cherché une réalisation de l'opérateur sous la forme d'un filtre à réponse impulsionnelle infinie (RII)

1.4.3 Amincissement des lignes de contours

Lorsque l'on effectue le gradient d'une image, il est nécessaire d'isoler les maximums locaux de l'image dérivée pour déterminer les points exacts de contour afin de réduire ceux-ci à une courbe d'un seul pixel d'épaisseur. Une des techniques les plus simples consiste à remettre à zéro toutes les valeurs de gradient qui ne sont pas maximales par rapport à celles des pixels voisins situés dans la direction du gradient. En effet, la valeur du pixel doit être maximale localement à la position exacte du contour et décroître de part et d'autre de celui-ci dans la direction du gradient. Soit la méthode suivante dite « suppression des points non maximum », dont le principe est le suivant [22, 23] :

Soit $A(x,y)$ l'image composée des amplitudes des gradients, et $Dir(x,y)$ l'image des directions. Pour chaque point $A(x,y)$, on détermine les points adjacents qui se trouvent dans la direction du gradient, comme illustré sur la figure 4. Soient $A(x_1,y_1)$ et $A(x_2,y_2)$ ces points.

L'amincissement s'effectue comme suit : si $A(x,y)$ est supérieur à la fois à $A(x_1,y_1)$ et à $A(x_2,y_2)$, alors $A(x,y)$ est conservé sinon $A(x,y)$ est mis à zéro.

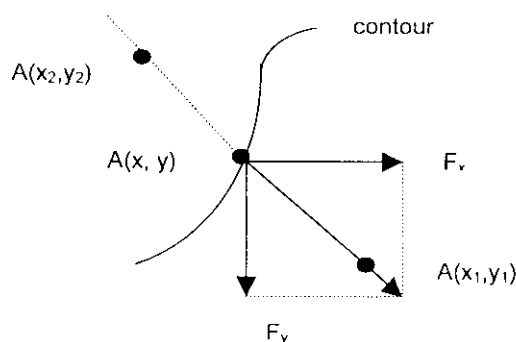


Figure 4 : Exemple d'amincissement.

Cette technique bien qu'utilisée pour de nombreux détecteurs, elle est à l'origine des trous situés près des jonctions regroupant plusieurs contours [23], d'où les contours obtenus sont incomplets. Il existe d'autres techniques d'affinage [22], nous présentons au chapitre suivant une méthode basée sur la morphologie mathématique.

1.4.4 Seuillage

Pour obtenir l'image des contours, nous devons binariser l'image dérivée et amincie (du gradient ou du laplacien). Le seuillage est l'opération qui associe à chaque point de l'image une classe de luminance après comparaison de ses attributs (niveaux de gris) à un ou plusieurs seuils [4]. Une étiquette² e est alors affectée à chacun des points en fonction de la classe d'appartenance.

Un seuillage classique s'avère en général insuffisant. De faux pixels-contours sont détectés dans les zones bruitées ou les zones de texture (même légèrement) et des pixels contours importants dont le gradient est faible sont oubliés. Plusieurs techniques de seuillage [1, 4] existent, le seuillage par hystérésis est la technique la plus répandue. Cette technique consiste à choisir deux seuils S_1 et S_2 ($S_1 > S_2$) et à effectuer un chaînage en opérant une sélection des gradients de la façon suivante [22, 23] :

L'image est parcourue par exemple de haut en bas et de gauche à droite. Dès qu'une valeur de gradient dépasse S_1 , elle constitue le départ de la chaîne. On essaie alors de continuer la chaîne dans les deux sens, en fonction de l'orientation du gradient. Tant que le gradient sur le pixel suivant est supérieur à S_2 , on continue la chaîne, sinon une extrémité de la chaîne est atteinte. Lorsqu'une chaîne est terminée, le parcours de l'image reprend là où il s'était arrêté. Tous les pixels chaînés sont bien sûr marqués pour qu'ils ne soient pas traités plusieurs fois.

1.4.5 Le prolongement, la correction et le chaînage

Un prolongement des contours peut s'avérer nécessaire pour deux raisons. Premièrement, les résultats du détecteur ne sont pas toujours de grande qualité et de nombreux contours tronqués doivent être rallongés et raccordés. Deuxièmement, pour certaines applications, il est intéressant d'avoir des contours fermés pour travailler ensuite sur les formes obtenues.

Il existe de très nombreuses techniques pour prolonger les contours. La plupart d'entre elles sont fondées sur la recherche d'un parcours optimal dans un graphe évalué. De façon générale, il s'agit d'exploiter les informations sur le gradient, associant intensité et direction, ainsi que l'orientation à l'extrémité pour générer des prolongements jusqu'à ce qu'un critère d'arrêt soit vérifié, comme par exemple la proximité d'un autre contour, un gradient trop faible ou un nombre limite de pixels-contours ajoutés.

Il est aussi possible de raisonner uniquement sur les chaînes de pixels-contours ; si deux contours sont suffisamment proches l'un de l'autre et à peu près alignés, ils sont prolongés et raccordés.

Le prolongement des contours est souvent considéré comme une étape de correction [23], puisque les informations utilisées pour rallonger les contours restent essentiellement les mêmes que pour la détection des contours, c.-à-d., l'intensité et la direction du gradient.

² Exemple : Un point peut être classé contour ou non-contour.

Par la figure suivante, nous présentons l'application de certains des détecteurs déjà présentés sur une image³ d'un objet de forme rectiligne. En dessous de chaque image résultante de la détection de contours, l'opérateur et le seuil S sont donnés. Notons que les seuils utilisés sont trouvés empiriquement.

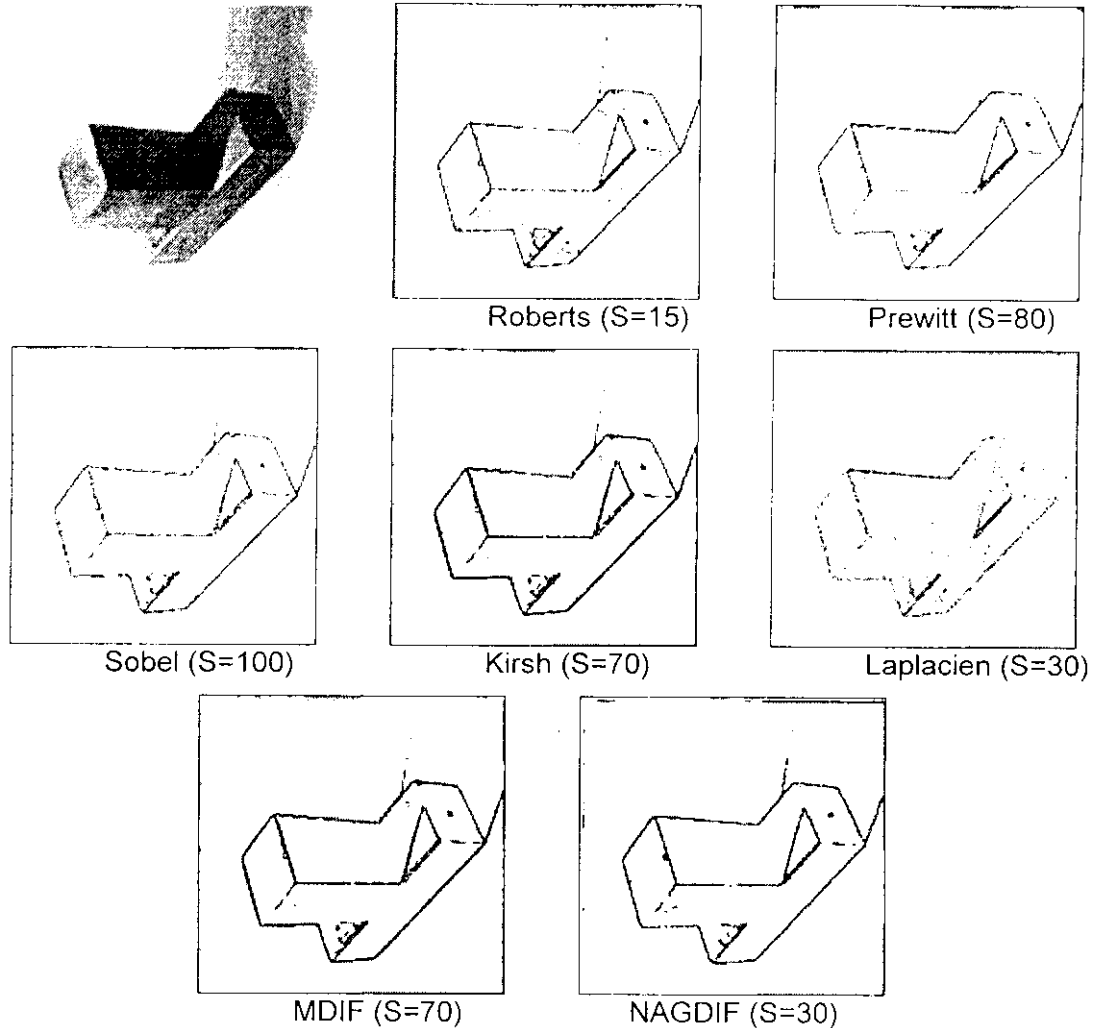


Figure 5 : Application de différents détecteurs de contours.

Les détecteurs sont appliqués sur plusieurs images de scènes réelles, par la figure 5, nous avons présenté qu'un échantillon.

Nous pouvons avancer prudemment quelques observations :

- ✓ Les opérateurs du premier ordre sont particulièrement sensibles au bruit. Contrairement à l'opérateur de Roberts, les opérateurs Sobel et Prewitt présentent l'avantage d'être moins sensibles au bruit, puisque la sensibilité de la dérivation a été diminuée par le calcul de la moyenne locale sur le domaine couvert par le masque.

³ Dans le cadre de cette thèse, toutes les images utilisées sont à plusieurs niveaux de gris, de type bmp, de taille 256x256 codées sur 8 bits.

- ✓ Avec les opérateurs de dérivées secondes, les contours sont moins réguliers que ceux fournis par les opérateurs du premier ordre et la sensibilité au bruit est plus importante, puisque le calcul d'une dérivée seconde de l'image est très instable.
- ✓ Les opérateurs NAGDIF et MDIF présentent de bons résultats malgré leur simplicité. Il faut signaler que les contours obtenus avec MDIF résultent d'une extraction des maximums locaux dans l'image de la norme du gradient. Les deux opérateurs sont intéressants et ont l'avantage d'être rapides en temps de calcul.

1.5 Approches régions

Dans ce second type d'approches, on ne suppose plus comme dans la première famille que l'on peut trouver des contours autour des différents objets que l'on veut mettre en évidence. On cherche plutôt à trouver dans quelle région de l'image ces objets se trouvent. On s'axe donc sur les régions.

La segmentation en régions est un des problèmes les plus difficiles de la vision par ordinateur. Elle consiste à décrire l'image à l'aide de régions pertinentes regroupant tous les pixels respectant une même propriété. Certaines applications sont fondées sur une segmentation où chaque région correspond à une partie bien précise d'un objet [7, 8, 33]. Le problème est alors ramené à une mise en correspondance directe entre les régions et les objets de la scène. La segmentation est donc une étape très importante, pratiquement incontournable pour de nombreuses applications.

Nous pouvons classer les méthodes de segmentation régions en deux grandes catégories [22] : Celles qui exploitent des connaissances sur les objets pour aider la segmentation et celles qui ne les exploitent pas

Dans les paragraphes suivants, nous présentons un état de l'art sur les techniques de segmentation en régions homogènes.

1.5.1 Méthodes basées sur l'exploitation de l'histogramme

Il existe des méthodes de segmentation simples pour obtenir des résultats très performants dans les applications où les objets ont une couleur ou une intensité lumineuse uniforme et bien distincte des autres objets et du fond. Ces méthodes exploitent l'histogramme des niveaux de gris. En effet, lorsque les objets se détachent du fond, l'histogramme comporte plusieurs blocs, appelés modes, correspondant à chaque objet. Le principe de base est de détecter les vallées entre chaque mode pour obtenir les seuils sur les niveaux de gris qui segmentent l'image en régions de façon satisfaisante [2, 4, 22]. Il existe de très nombreuses variantes qui sont classées en deux grandes catégories que nous décrivons brièvement ci-dessous.

La première catégorie regroupe toutes les techniques qui calculent des seuils globaux appliqués à toute l'image. La classification de Fisher détermine le ou les seuils qui minimisent la somme des variances de niveaux de gris de chaque classe, le nombre de modes étant fixé par l'utilisateur.

Pour améliorer la forme de l'histogramme et faciliter le choix du seuil, certaines techniques [1, 2, 4] sélectionnent les pixels qui vont être pris en compte dans le calcul de l'histogramme. Il existe deux approches possibles, celles qui tentent de ne pas tenir compte des pixels situés à la frontière entre les objets et le fond, et celles au contraire qui tentent de ne prendre en compte que ces pixels, ou une partie d'entre eux.

Weszka, Nagel et Rosenfeld [4] proposent ainsi de sélectionner les pixels qui ont une forte valeur de Laplacien. Ceci entraîne normalement un histogramme bimodal équilibré car la valeur du Laplacien est forte juste au début et à la fin d'une forte transition mais pas au milieu d'une région homogène, ni au milieu de la transition. Le seuil choisi est donc au creux de l'histogramme.

La deuxième catégorie regroupe les techniques qui calculent des histogrammes locaux. Pour la plupart de ces méthodes classiques de seuillage local, l'image est subdivisée en petites sous-images et ensuite les techniques globales de seuillage sont fréquemment utilisées. Nous en citons celles-ci, Chow et Kanelo [4] proposent de calculer la variance et l'histogramme des valeurs de gris pour un ensemble de fenêtres de l'image. Si la variance dépasse un certain seuil et si l'histogramme est plutôt bimodal qu'unimodal, la fenêtre est partagée en fonction du seuil choisi dans la vallée de celui-ci. Pour les pixels n'appartenant à aucune fenêtre, la classification est déterminée par interpolation des seuils trouvés dans les fenêtres voisines.

Watanabe [4] suggère de choisir une valeur du seuil qui maximise la somme des gradients de tous les pixels dont le niveau de gris est égal à la valeur du seuil.

1.5.2 Croissances de régions

Contrairement aux méthodes de seuillage qui ne tiennent pas compte des relations spatiales des pixels. La segmentation par croissance de régions utilise aussi bien les relations spatiales entre les points que leurs propriétés. Le problème fondamental associé à cette catégorie d'approches est justement lié à la définition du critère d'homogénéité utilisé dans le processus de partitionnement de l'image.

Dans cette catégorie, il existe trois grandes familles de méthodes : de division, de fusion et de division-fusion. Là encore, il existe de très nombreuses méthodes, nous citons les grands principes en détaillant quelques approches.

L'idée de la fusion consiste à exploiter une partition initiale de l'image constituée de petites régions. Puis ces régions sont fusionnées successivement jusqu'à ce que le critère de fusion ne soit plus vérifié.

La division consiste à partitionner l'image qui constitue la région initiale en régions de plus en plus homogènes. Le processus est réitéré pour chacune des régions produites jusqu'à ce qu'une certaine homogénéité soit atteinte. Cette homogénéité est souvent contrôlée par sa variance ou son contraste.

Les méthodes de division-fusion combinent les deux approches précédentes. La division qui partitionne l'image en zones localement homogènes, puis la fusion des régions similaires au sens d'un prédicat de regroupement.

Une des premières idées pour regrouper les pixels a été de comparer leurs niveaux de gris et de lier tous ceux dont la différence est inférieure à un seuil. Pour améliorer la robustesse, il est préférable de prendre en compte une partie du voisinage de chaque pixel. L'inconvénient de cette technique est son caractère uniquement local, en effet, la connexité entraîne parfois la présence dans la même région de pixels ayant des propriétés très différentes [1].

Les techniques de fusion et de division des régions permettent la prise en compte d'informations plus globales, selon des critères plus ou moins complexes faisant intervenir par exemple la différence de moyenne des deux régions, leurs variances, leur nombre de pixels et le nombre de pixels adjacents.

Dans ce qui suit, nous présentons en détails quelques algorithmes de croissance de régions.

En commençant par un petit nombre de pixels connexes, cette zone est itérativement fusionnée à d'autres petites régions avoisinantes. Horowitz et Pavlidis ont été les précurseurs qui ont suggéré ce point de vue. Ils proposent un algorithme récursif de division de l'image par un parcours arborescent (quadtree), suivi par une technique de fusion de régions [1, 23]. En premier lieu, la région est découpée en 4 sous-régions carrées, et ainsi de suite récursivement jusqu'au niveau du pixel. Ensuite, à chaque fois que 4 sous-régions satisfont un critère d'homogénéité, elles sont regroupées à un niveau supérieur du quadtree. La fusion continue tant qu'il est possible de la faire, c.-à-d., tant que le critère d'homogénéité est satisfait. Lorsqu'il n'est plus possible de fusionner, il reste encore une étape pour examiner les régions adjacentes qui n'étaient pas au même niveau dans le quadtree et les fusionner si elles satisferont les critères d'homogénéité.

Cette méthode laisse une empreinte spécifique dans la carte des régions avec des frontières marquées par de nombreux traits verticaux et horizontaux, ce qui occasionne parfois des régions particulièrement découpées.

Le point de contrôle de cet algorithme consiste à définir le prédicat d'homogénéité, qui décide si une région doit être divisée ou pas, ainsi que le critère de similarité, qui choisit le meilleur couple de régions voisines à fusionner. En effet, le critère d'homogénéité joue un rôle très important dans la fusion des régions mais celui-ci n'a pas fait l'objet d'une évaluation, ce qui est dommage. L'intérêt de cette technique est avant tout la rapidité et la taille mémoire relativement faible qui est nécessaire pour segmenter l'image.

Ohlander et al [34] proposent une division récursive de l'image. Etant donné une région connexe, l'histogramme est calculé pour chaque couleur ou intensité. Un lissage de l'histogramme est effectué, puis le meilleur pic est sélectionné. Ce pic est censé déterminer la moyenne des valeurs d'une région pour une intensité donnée. Deux solutions se présentent. Si

les limites de cette région apparaissent clairement, c.-à-d., si des vallées importantes sont trouvées de part et d'autre de ce pic, alors ces vallées correspondent aux seuils choisis pour identifier tous les pixels de cette région. Sinon la région est correctement segmentée, il ne faut plus y toucher. A chaque fois qu'une décomposition a lieu, les régions résultantes sont traitées pour que les trous soient éliminés, puis les composantes connexes des régions formées sont réexaminées par le même principe de façon récursive.

Adams et Bishof [36] ont développé une approche qui repose sur la construction de la région pixel par pixel, car ainsi les caractéristiques isolées d'un point n'avantagent pas ou ne pénalisent pas le groupe de points auquel il est artificiellement rattaché. En effet, le démarrage se fait sur un ensemble de germes de régions, tous les pixels candidats situés dans le voisinage des régions courantes sont classés dans une liste commune. Le meilleur pixel est ajouté en premier à la région dont il est voisin. Les pixels voisins de plusieurs régions apparaissent deux fois dans la liste, puisqu'ils sont évalués par rapport à chacune des régions qu'ils bornent. L'utilisation d'une liste unique de pixels candidats assure une croissance de région qui se fera en parallèle, dans laquelle les pixels sont fusionnés de façon concurrente aux régions pour lesquelles ils obtiennent la meilleure évaluation. Naturellement, les régions homogènes seront donc naturellement privilégiées par ce mécanisme.

1.5.3 Segmentation par relaxation

Les techniques de relaxation sont caractérisées par le fait qu'un ensemble de processus locaux entrent en interaction à travers des coopérations et des compétitions afin de conduire à un consensus final quand à l'interprétation de l'image [37].

Contrairement aux techniques présentées jusqu'à présent, les techniques de relaxation génèrent une suite de segmentation au cours de laquelle les ambiguïtés et les incohérences entre labels affectés à des sites voisins sont progressivement réduites. La relaxation probabiliste et la relaxation stochastique constituent deux approches distinctes. Elles ont cependant ceci en commun que l'étiquetage d'un site dépend du contexte dans lequel il est plongé [15]. Le contexte est représenté par les labels de ses voisins immédiats. En conséquence, les calculs impliqués dans la mise en œuvre de ces méthodes sont purement locaux.

La relaxation probabiliste a pour objectif de déterminer à partir d'une segmentation initiale, une segmentation dans laquelle les labels attribués à des sites voisins satisfont des contraintes de compatibilité mutuelle. Le principe général en est d'attribuer à chaque site un vecteur de poids qui expriment les préférences en ce site pour chacun des labels possibles. Les contraintes sont représentées par des coefficients de compatibilité entre labels de sites voisins. Par relaxation, les poids sont itérativement modifiés et se concentrent progressivement sur les labels les plus compatibles. En dépit du nom qui lui a été attribué, la relaxation probabiliste est une méthode purement déterministe dans laquelle chacun des poids ne peut varier que dans un

sens. L'intérêt de cette technique est de permettre l'amélioration d'un étiquetage des données obtenu au préalable par une autre technique.

La relaxation probabiliste vise à déterminer un étiquetage des sites qui satisfait des contraintes sur les labels. La relaxation stochastique constitue une généralisation de cette approche, dans laquelle la segmentation recherchée doit satisfaire des contraintes qui concernent à la fois les données et les labels [15, 39, 40].

1.6 Segmentation des images texturées

La segmentation par la texture constitue un cas particulier qui nécessite le recours à des descripteurs de texture. Il n'existe pas de définition unique pour la texture [41, 42]. La définition précise de celle-ci n'est toujours pas donnée. La difficulté principale provient du fait qu'elle est étroitement liée à la perception visuelle humaine et par conséquent contient une composante subjective se prêtant mal à une formalisation. Il est cependant communément admis que la texture correspond à l'arrangement spatial des intensités lumineuses. Plus que les valeurs absolues des intensités, ce sont donc les valeurs relatives entre pixels voisins qui portent l'information caractéristique d'une texture donnée.

La texture est une région de l'image dont l'observation se traduit par une impression visuelle d'homogénéité pour toutes les transitions possibles à l'intérieure de cette région. Elle traduit une répartition spatiale d'un même motif dans différentes directions de l'espace. Ceci lui attache donc deux dimensions sur la base desquelles on peut la décrire. La première concerne les motifs élémentaires constituant la texture. La seconde implique la distribution spatiale de ces motifs [41, 42, 43].

Un motif élémentaire est un ensemble connexe de points caractérisé par des propriétés de nature quantitative telles que les moments, l'entropie et l'énergie, ...

La répartition spatiale des motifs nécessite la définition d'une fonction permettant la prise en compte des intercorrélations ou d'un ensemble de règles de placement des motifs sur le support bidimensionnel. La texture est donc une propriété définie sur un certain domaine spatial. En conséquence, les descripteurs ponctuels ne sont pas valides pour la segmentation d'images texturées.

Les approches les plus fréquemment rencontrées dans la littérature pour l'analyse de texture peuvent être classées en deux grandes classes : L'approche statistique et l'approche structurelle [41].

Il existe deux catégories dans l'approche statistique dont :

1. Méthodes par extraction de paramètres et classification : Il s'agit d'extraction de mesures statistiques qui peuvent ensuite être utilisées comme attributs en entrée de schéma de classification et d'identification. Il s'agit ainsi de définir un vecteur d'attributs dont les composantes doivent être représentatives, pertinentes et discriminantes de façon qu'il puisse identifier sans ambiguïté une texture parmi d'autres. Les paramètres

sont donc déterminés suite à une analyse qui peut être faite de manière globale ou de manière locale.

Ils existent plusieurs paramètres de texture [41, 42] : les histogrammes en niveaux de gris, les matrices de co-occurrences, les longueurs de plage, paramètres morphologiques, fractals, ...

Les descripteurs statistiques sont utilisés dans la présentation des microtextures, les textures sont caractérisées par les fréquences d'apparition d'attributs locaux en chacun des points de la texture.

L'utilisation des matrices de co-occurrences comme descripteurs de texture repose sur l'idée qu'une texture est caractérisée par les fréquences avec lesquelles des niveaux de gris surviennent en des positions relatives données. On détermine ainsi, pour un certain nombre de déplacements relatif δ , et pour toutes les valeurs des niveaux de gris i et j , le nombre $m_{i,j}(\delta)$ de couples de pixels séparés par le déplacement δ et porteurs des valeurs i et j . Les résultats sont exprimés sous la forme d'une matrice de co-occurrences :

$$M(\delta) = (m_{i,j}(\delta))_{i,j} \quad (22)$$

δ est un vecteur qui correspond à une distance d et à une direction θ .

Les matrices de co-occurrences sont des descripteurs de texture dont la dimension est très importante. Celle-ci augmente en effet avec le nombre de niveaux de gris dans l'image pour obtenir un ensemble plus compact. On extrait en général d'autres descripteurs à partir de ces matrices, Haralick et al [42] ont ainsi initialement défini 14 fonctions de second ordre sur ces matrices. Souvent que 4 ou 5 qui sont retenues. Certaines des fonctions les plus utilisées sont l'énergie, l'entropie, le contraste, et la corrélation :

$$Energie = \sum_i \sum_j p_{i,j}^2 \quad (23)$$

$$Entropie = \sum_i \sum_j p_{i,j} \log p_{i,j} \quad (24)$$

$$Contraste = \sum_i \sum_j (i-j)^2 p_{i,j} \quad (25)$$

$$Corrélation = \sum_i \sum_j (ij p_{i,j} - \mu^2) / \sigma^2 \quad (26)$$

où $p_{i,j}$ est la valeur de $m_{i,j}(\delta)$ après normalisation, et μ et σ^2 sont la moyenne et la variance des lois marginales de i et j (les matrices étant symétriques, ces lois sont identiques). Il existe plusieurs autres descripteurs dans la littérature, tel que les différences de niveaux de gris [3, 22], paramètres morphologiques, paramètres fractals...

Dans le cas de l'approche morphologique [44, 45, 46], les formes analysées sont les formes des motifs constituant la texture. La génération d'une image binaire résulte de la translation d'un élément structurant au travers de l'image et application d'un opérateur morphologique par cet élément structurant.

Certaines études ont mis en évidence la correspondance entre la dimension fractale et l'évaluation humaine de la rugosité [47]. Ils existent plusieurs méthodes pour le calcul de cette dimension [48, 49, 50, 51, 52]. Souvent l'attribut de lacunarité est ajouté à la dimension fractale [47], qui est faible quand la texture est fine et forte pour une texture grossière.

2. Méthodes par modélisation de texture : Il existe deux types de modélisation [43].

La première est la modélisation linéaire de la texture, où la valeur du niveau de gris de chaque pixel est considérée comme une combinaison linéaire de celle des pixels voisins, différentes modélisations linéaires peuvent être employées pour décrire un signal, modèle Auto-Regressive AR, modèle Moving Average MA, le modèle ARMA qui est le couplage des deux modèles AR et MA, méthode d'auto-corrélation et méthode de corrélation.

La seconde est la modélisation par les champs de Markov. Les interactions locales aux niveaux des points sont prises en compte. La tâche principale est l'étiquetage, où à chaque pixel, on veut associer une étiquette appartenant à un ensemble fini. Le problème est de choisir une étiquette optimale pour un pixel. La segmentation associée aux champs de Markov et la théorie bayésienne [22], se ramène à un problème d'optimisation. Le but est alors de trouver le minimum global d'une fonction représentant l'erreur commise.

L'inconvénient de la première catégorie des méthodes est que les paramètres (vecteurs de déplacements, seuils, tailles de fenêtres) sont souvent difficiles à trouver et exigent des images de grandes tailles avec un nombre important de niveaux de gris et un temps de calcul très important.

L'inconvénient des méthodes de modélisation est qu'elles exigent un nombre considérable d'itérations pour converger vers des solutions stables, donc le temps de calcul est important. Et la performance s'avère néanmoins limitée dans le cas des textures à caractère régulier.

Les descripteurs structuraux [41], dans lesquels une texture est considérée comme un ensemble de primitives suivant les règles de placement, plus adaptés au cas des macrotextures. Cette approche cherche à mettre en évidence une structuration plus ou moins régulière des éléments de la texture et correspond à une vision plutôt macroscopique de celle-ci. Elle est particulièrement adaptée au cas où les motifs sont bien discernables. Les domaines d'application de l'approche structurale étant relativement limités.

1.7 Coopération par fusion région/ contour

Chaque méthode de segmentation seule présente des défauts intrinsèques. Les atouts de chaque méthode peuvent donc permettre de corriger leurs défauts. Plus précisément, la détection des contours peut aider à contrôler l'expansion ou la fusion des régions, tandis que la détection des régions contribue à la localisation et à la fermeture des contours. C'est l'idée de base qui sous entend les coopérations de méthodes.

De manière générale, la coopération se fait essentiellement des contours vers les régions, ceci est probablement dû à l'incapacité des détecteurs de contours à exploiter les informations globales véhiculées par les régions, et aussi, parce que les critères utilisés par les méthodes de croissance de régions sont souvent trop globaux, et les frontières délimitant ces objets ne sont généralement pas significatives. Notons que même si les contours sont remis en question, ils sont calculés au début de la coopération, sans l'aide des informations sur les régions.

Peu de travaux exploitent totalement la dualité entre ces deux types de primitives et la coopération se situe surtout au niveau de la confrontation des résultats, mais pas au niveau de la prise de décision de chaque processus.

Les problèmes majeurs sont l'intégration des informations complémentaires et le contrôle de la coopération qui a pour charge la gestion et l'enchaînement des étapes.

1.8 Conclusion

Après avoir présenté l'état de l'art sur la segmentation, nous énonçons que :

Le problème fondamental pour les méthodes de détection des contours est la résistance au bruit et aux petites variations locales dues à la texture ou à des irrégularités de surface. Les opérateurs du premier et du second ordre donnent des résultats satisfaisants dans le cas d'images simples. Ils sont néanmoins parfois moins précis pour la localisation des contours dans des images complexes et engendrent souvent de fausses détections.

Les techniques basées sur l'exploitation de l'histogramme exploitent exclusivement les informations de niveau de gris sans tenir compte de la position des pixels (à quelques exceptions près), ce qui marque les faiblesses de ces approches, qui restent toutefois intéressantes de par leur simplicité et leur efficacité pour des applications simples. L'avantage principal des méthodes de seuillage réside dans le fait que des pixels quelconques de même niveau de gris sont traités d'une manière identique. Il n'est pas alors nécessaire de classer tous les pixels d'une image, mais simplement les différentes valeurs de ses niveaux de gris.

Les méthodes de fusion sont très coûteuses en temps de calcul, en effet, la fusion implique à chaque itération, une recherche sur toute l'image des deux régions les plus aptes à fusionner.

L'inconvénient des méthodes de division est qu'elles reposent sur des statistiques globales, une petite région située à côté d'une grande sera donc rarement restituée. De plus, de par le principe même de la division, les images sont souvent sur-segmentées.

La division-fusion permet de garantir une segmentation qui vérifie les propriétés d'homogénéité imposées, néanmoins elles sont peu robustes et sensibles aux conditions de

départ (la partition initiale, le ou les seuils de tolérance de fusion et/ ou division ainsi que le sens du parcours de l'image). Une légère modification de l'un de ces facteurs peut conduire à des segmentations très différentes. Un grand avantage d'une méthode de division-fusion est qu'elle conduit à des algorithmes facilement parallélisables : toute une partie du traitement (la division) peut être réalisée de manière séparée et peut donc être partagée sur différents processeurs.

Le critère idéal de fusion pour les méthodes de croissance de régions dépend théoriquement de très nombreuses informations comme, entre autres, la moyenne et l'écart type des niveaux de gris, ... Notons que toutes les régions d'une image respectent des propriétés très différentes, souvent insuffisantes à exprimer ou à mesurer avec la moyenne ou l'écart type, ...

Du fait de la nature contextuelle et globale de l'analyse des méthodes de segmentation de textures, elles souffrent dans leur majorité du problème de mauvaise localisation des frontières des régions. Elles présentent en outre le défaut d'altérer les structures fines dans l'image qui sont souvent soit fusionnées aux régions voisines soit sur-segmentées.

L'évaluation de la segmentation demeure le problème le plus délicat, puisque souvent elle se fait par comparaison entre la segmentation obtenue et une carte de référence définie par l'image. Cette dernière n'existe pas ou est très complexe à avoir surtout quand il s'agit d'images réelles telles que les images médicales.

Dans la suite de notre travail, nous présentons plusieurs autres méthodes de segmentation basées sur différents principes et théories.

Chapitre 2

Segmentation des Images : Algorithmes et Applications

- 2.1 Introduction
- 2.2 Approches de détection de contours
- 2.3 Approches de segmentation en régions homogènes
- 2.4 Segmentation par étiquetage en composantes connexes
- 2.5 Conclusion

Segmentation des Images :

Algorithmes et Applications

2.1 Introduction

Beaucoup de travaux ont été consacrés au problème de la segmentation. Chronologiquement, ce sont les méthodes adaptées aux images uniformes au sens du niveau de gris et faiblement bruitées qui sont apparues les premières. Plusieurs méthodes d'analyse de textures ont ensuite permis de segmenter des images texturées.

Ce chapitre est une continuité du précédent, nous allons présenter, étudier et implémenter des techniques basées sur des principes différents.

En effet, en plus des algorithmes basés sur le calcul du gradient, du laplacien et utilisant le filtrage optimal pour la détection de contours déjà vus, d'autres approches basées sur des théories plus complexes seront données, telles que les méthodes basées sur le calcul de la dimension fractale locale, ou encore la recherche des discontinuités selon un seuillage par rapport à la dimension fractale sur l'espace des exposants de singularité. Ces détecteurs seront appliqués sur différentes images de scènes réelles, quelques résultats seront présentés. Nous mettrons en évidence les inconvénients et les avantages de chaque méthode.

Pour la segmentation en régions homogènes, nous allons présenter et développer des techniques basées sur différentes théories telles que : multifractale, morphologie mathématique, logique floue, ... Elles seront appliquées et testées sur plusieurs types d'images.

Nous considérerons aussi la difficulté ou non de trouver les valeurs adéquates des paramètres d'un algorithme. Des solutions seront proposées pour simplifier leurs recherches.

La segmentation par classification de régions sera aussi étudiée, elle trouve son intérêt non pas dans l'extraction des classes, mais dans l'étiquetage optimal des points de l'image.

Une présentation des différents algorithmes de segmentation étudiés et des résultats d'application sur des images de scènes réelles seront donnés dans les sections suivantes.

2.2 Approches de détection de contours

2.2.1 Détection de contours par la théorie fractale

Les paramètres fractals et multifractals sont utilisés dans une approche de l'analyse d'images qui exploite une certaine «homogénéité» de groupements connexes de pixels. Cette approche est duale à la recherche des contours qui s'intéresse aux «hétérogénéités» dans l'image ; et comme nous le verrons, le calcul de la dimension fractale ou des spectres de

singularités est une étape dans la détection de contours et de segmentation en régions homogènes.

Une des caractéristiques principales de tout objet fractal¹ est sa dimension fractale, qui mesure son degré d'irrégularité et de brisure. Fait très important, la dimension fractale peut très bien être une fonction simple et même un nombre irrationnel.

La dimension fractale d'une image est comprise entre 2 et 3. Elle mesure le degré d'irrégularité et de rugosité de la surface des niveaux de gris dans un espace 3D ; plus la texture est lisse (respectivement rugueuse), plus la dimension fractale est proche de 2 (respectivement 3). La dimension fractale ne suffit en général pas à caractériser une texture. Il faut souvent lui ajouter le paramètre de lacunarité [47], qui est un paramètre fractal du second ordre mesurant la géométrie d'un ensemble de pixels par rapport au reste de l'image.

Dans les images de dimensions fractales, c.-à-d., chaque point (i,j) est associé à sa dimension fractale, les contours sont rehaussés. Ceci est expliqué par le fait que les limites entre les régions homogènes sont une configuration physique qui ne correspond pas bien au modèle fractal. Ainsi quand les points qui appartiennent à ces limites entre deux régions de l'image sont examinés, c.-à-d., points contours, l'approximation entre le modèle fractal et les données de l'image est faible. Le fait, que les limites soient plus élevées que les surfaces d'intensités non fractales, fournit le moyen de détecter les points de l'image qui peuvent bien être des contours. Il découle que quand la dimension fractale d'une région couvrant la limite entre deux régions homogènes est calculée, la valeur obtenue est inférieure à la dimension topologique, c'est à dire 2. Donc, quand les valeurs des dimensions fractales sont quantifiées en niveaux de gris, les contours seront vus en noir et le reste des régions aura une nuance de gris particulière.

Cette propriété des dimensions fractales est exploitée, ce qui a permis de développer différents algorithmes [47, 48, 49, 50, 57, 58, 59], tel que la méthode de Blanket, la méthode basée sur le modèle du mouvement brownien fractionnaire FBM² ainsi que la méthode vario-spectrale. Cette dernière sera présentée dans ce qui suit.

2.2.1.1 Algorithme Spectral [59]

Le spectre de puissance d'une surface fractale $P(k_x, k_y)$ est donné par la relation suivante :

$$P(k_x, k_y) = \frac{1}{((k_x^2 + k_y^2)^{1/2})^H} = \frac{1}{\rho^H} \quad (1)$$

où $H > 0$,

k_x, k_y : les deux coordonnées fréquentielles selon les directions x et y respectivement,

ρ : la fréquence radiale, égale à : $(k_x^2 + k_y^2)^{1/2}$,

¹ Fractale est une figure géométrique de structure complexe dont la création ou la forme met en jeu des règles utilisant le fractionnement [55, 56]. Cette construction est liée à l'invariance d'échelle où on répète la même transformation à des échelles de plus en plus petites

² FBM est une extension du concept du mouvement Brownien [56, 61], ce type de mouvement est apparenté à une marche aléatoire, c.-à-d., une séquence d'étapes dont la taille et la direction sont déterminées par le hasard.

θ : la fréquence angulaire $\theta = \arctan(k_y / k_x)$,

Puisqu'il est admis que les images approximent les surfaces fractales, l'équation précédente est utilisée pour calculer la dimension fractale de l'image. En effet, l'exposant β est en relation avec la dimension fractale D :

$$\beta = 8 - 2D \quad (2)$$

Ainsi, la dimension fractale D peut être calculée à partir de l'exposant $-\beta$ de l'équation du spectre de puissance. Il suffit simplement de représenter le spectre de puissance suivant une échelle logarithmique ; alors la courbe en loi de puissance devient une ligne droite avec une pente équivalente à $-\beta$. Pour une image, la procédure donne une valeur globale de la dimension fractale. Mais dans notre cas, on s'intéresse à l'estimation de la dimension locale, c.-à-d., en chaque pixel de l'image.

Pour échantillonner le spectre de puissance à différentes fréquences pour chaque point de l'image, une méthode basée sur la convolution de l'image par ce filtre a été proposée [59].

$$h_{k_x, k_y}(x, y) = \frac{1}{\sqrt{\pi}\sigma} \exp\left[-(x^2 + y^2) / 2\sigma^2\right] \exp\left[2\pi j(k_x x + k_y y)\right] \quad (3)$$

où x, y : les coordonnées spatiales.

k_x, k_y : les coordonnées du plan des fréquences.

L'algorithme est décrit comme suit :

1. Choisir N filtres $h_{\rho_1, \theta}, h_{\rho_2, \theta}, \dots, h_{\rho_N, \theta}$ avec θ fixé.
2. Convolver l'image avec les N filtres, pour échantillonner le spectre de puissance à N fréquences pour chaque point de l'image.
3. Prendre le carré de l'amplitude des sorties des filtres. Ceci donnera N points pour chaque point de l'image.
4. Trouver la courbe en loi de puissance qui approxime le mieux les N points, pour chaque point de l'image. Exprimer l'exposant de la courbe de la meilleure approximation linéaire par $-\beta(x, y)$.
5. Calculer $D(x, y)$ pour chaque point de l'image en utilisant (2).

Notons qu'en utilisant un σ fixe, tous les filtres h_{k_x, k_y} auront la même largeur. L'échange entre les localisations fréquentielle et spatiale est déterminé par le choix de σ . On peut le voir par la figure 1³. Il est à noter que quand σ décroît, la localisation dans le plan (x, y) se précise.

³ Nous avons choisi une image constituée d'objets de formes rectilignes pour mieux apparaître les caractéristiques de l'algorithme

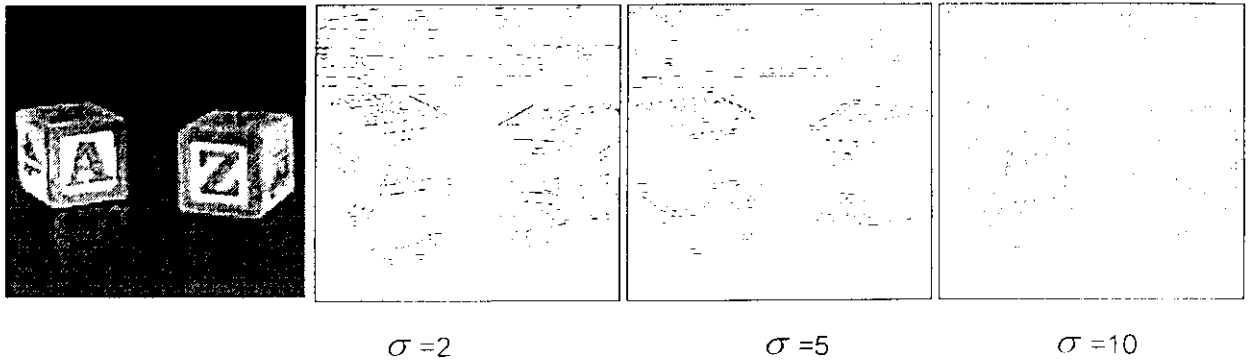


Figure 1: Localisation des contours pour différentes valeurs de σ et les fréquences radiales $\rho_1=20, \rho_2=20.2, \rho_3=20.5, \rho_4=20.7, \rho_5=21$ à $\theta=0=90^\circ$.

En appliquant l'algorithme sur la même image et mêmes paramètres d'entrée mais pour trois valeurs différentes des fréquences angulaires θ , on trouve les résultats de la figure 2.

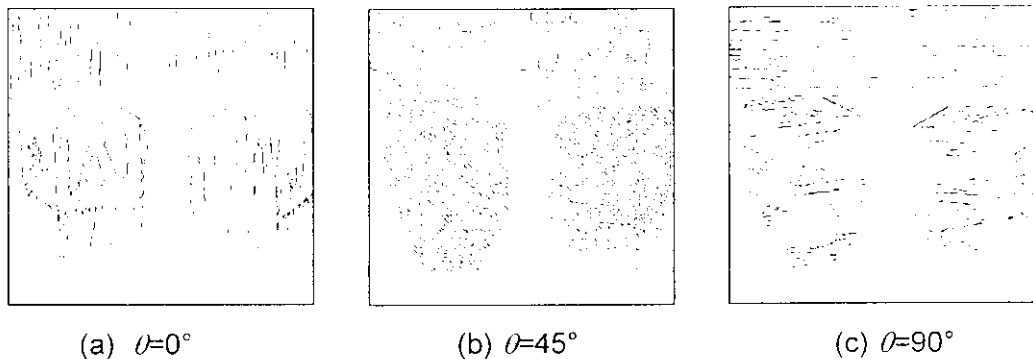


Figure 2 : Détection de contours pour différentes valeurs de θ avec $\sigma=2$ et les fréquences radiales : $\rho_1=20, \rho_2=20.2, \rho_3=20.5, \rho_4=20.7, \rho_5=21$.

Ces résultats montrent la caractéristique importante de cet algorithme à savoir son caractère isotropique [59, 60]. En effet, le spectre de puissance est anisotropique, puisque les mesures de la dimension fractale dans différentes directions sont différentes. Une seconde application de cet algorithme consisterait donc à l'utiliser comme extracteur sélectif de contours dans une direction donnée. Ce qui rend le choix du paramètre θ très important, et pour une meilleure détection de contours, il faudra appliquer l'algorithme sur un grand nombre de fréquences angulaires θ , mais le coût de calcul devient ainsi très important.

Nous avons cherché à utiliser d'autres moyens qui nous guideront à trouver la bonne fréquence angulaire pour une meilleure détection. Nous avons ainsi développé un algorithme basé sur l'exploitation d'indices locaux permettant d'affirmer l'existence de contours dans l'image par l'analyse des variogrammes, nous avons associé cet algorithme à la méthode spectrale, nous obtenons ainsi la méthode "vario-spectrale" [5]. La théorie et l'application de cette méthode sont présentées dans la section suivante.

2.2.1.2 Calcul du Variogramme

Le variogramme est introduit comme étant une fonction répondant à priori aux qualités requises pour être un bon indice de texture ; les variogrammes possèdent plusieurs propriétés théoriques et expérimentales [62]. Dans notre étude, la propriété du variogramme exploitée est la suivante :

« La direction privilégiée d'un motif ou d'un objet dans une image est celle dont l'amplitude du variogramme autour de l'origine est la plus faible »

Donc, pour détecter l'orientation des objets présents dans une scène, il suffit de calculer les courbes des variogrammes (l'axe des abscisses représente la valeur de la distance h et l'axe des ordonnées représente l'amplitude du variogramme) pour différentes directions, et par comparaison des amplitudes autour de l'origine, nous pouvons déduire la direction privilégiée.

Sachant que l'expression du variogramme d'un domaine d'image D_Ω de taille $N \times N$ dans la direction θ selon une distance h est la suivante :

$$\gamma_\theta(h) = \frac{1}{(D_\Omega)} \sum_{x=m+1}^N \sum_{y=m_2+1}^N (A(x,y) - A(x-m, y-m_2))^2 \quad (4)$$

où A représente l'image en niveau de gris et $N \times N$ la taille de l'imagette.

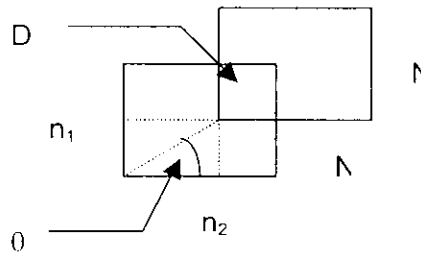


Figure 3 : Schéma d'un champ de mesure du variogramme

Ainsi, les expressions des variogrammes pour chaque direction θ sont données par :

$$\theta=0^\circ : \quad \gamma_H(h) = \frac{1}{(N-h)N} \sum_{x=h+1}^N \sum_{y=1}^N (A(x,y) - A(x-h,y))^2 \quad (5)$$

$$\theta=45^\circ : \quad \gamma_{DP}(h) = \frac{1}{(N-h)^2} \sum_{x=h+1}^N \sum_{y=h+1}^N (A(x,y) - A(x-h,y-h))^2 \quad (6)$$

$$\theta=90^\circ : \quad \gamma_V(h) = \frac{1}{(N-h)N} \sum_{x=1}^N \sum_{y=h+1}^N (A(x,y) - A(x,y-h))^2 \quad (7)$$

$$\theta=135^\circ : \quad \gamma_{DS}(h) = \frac{1}{(N-h)^2} \sum_{x=N-h+1}^N \sum_{y=h+1}^N (A(x,y) - A(N-x-h,y-h))^2 \quad (8)$$

2.2.1.3 Méthode Vario-Spectrale [5]

Pour l'estimation des fréquences angulaires θ nous proposons cette procédure :

- Nous divisons l'image originale en plusieurs sous images de taille $N \times N$.
- Sur chaque sous image, nous calculons les variogrammes de ces quatre directions $\theta = 0^\circ, 45^\circ, 90^\circ$ et 135° .

- Pour chaque sous image, nous cherchons la direction privilégiée. Ce qui nous donne une matrice de directions.
- Nous calculons l'histogramme des directions sur toute l'image (c.-à-d., La fonction qui donne la fréquence d'apparition de chaque direction sur toute l'image).
- La direction privilégiée de l'image correspond à celle ayant la plus grande fréquence.

Si nous souhaitons extraire les contours pour deux directions de θ , alors la deuxième direction privilégiée est celle ayant un histogramme le plus proche de la plus grande fréquence.

Après l'estimation des fréquences angulaires, nous procédons par la détection de contours selon l'algorithme spectral.

Pour l'image 2 dès, les deux directions privilégiées estimées sont $\theta=90^\circ$ et 0° , le résultat de détection de contours est donné par la figure suivante :

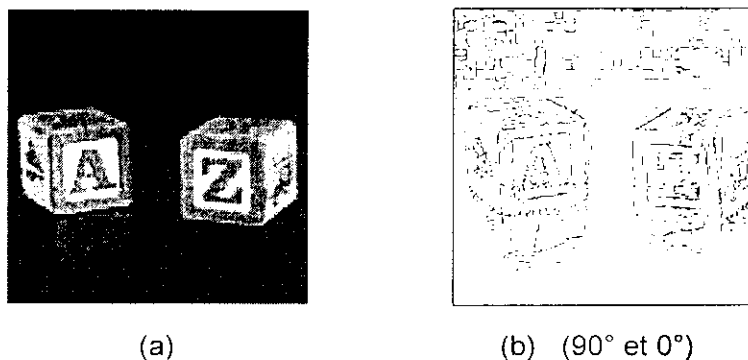


Figure 4 : Détection de contours par la méthode vario-spectrale pour deux valeurs de θ .

Notons qu'il est possible d'appliquer sur chaque imagerie sa direction privilégiée estimée, cependant, nous obtenons des contours interrompus sur toute l'image.

Malgré les améliorations apportées à la méthode spectrale par l'utilisation du variogramme [5], le temps d'exécution reste long et le choix des fréquences radiales reste toujours difficile.

La qualité de détection des contours est moyenne, nous observons souvent des contours interrompus et aussi des contours non détectés.

La dimension fractale globale de l'image peut être obtenue en faisant la moyenne des dimensions fractales locales de tous les pixels de l'image. Elle est exploitée dans la détection de la malignité ou de la bénignité des lésions dans des images mammographiques [59].

2.2.2 Détection de contours par la théorie multifractale

Avant de présenter l'algorithme, nous devons présenter des définitions [61] nécessaires pour faciliter sa compréhension.

A l'instar de la dimension fractale pour les ensembles irréguliers, le spectre $f(\alpha)$ des singularités peut être introduit comme une quantité qui donne une caractérisation relativement précise du degré de régularité et d'homogénéité d'une mesure fractale.

Exposant de singularité : Soit μ une mesure quelconque et $supp\mu$ son support. Une mesure permet d'associer des poids relatifs aux différentes parties d'un ensemble, son support est l'ensemble où la mesure est définie.

On appelle exposant de singularité au point $x_0 \in supp\mu$, la limite :

$$\alpha(x_0) = \lim_{\varepsilon \rightarrow 0^+} \frac{\ln \mu(B_{x_0}(\varepsilon))}{\ln \varepsilon} \quad (9)$$

où $B_{x_0}(\varepsilon)$ désigne une boule centrée en x_0 et de taille ε .

L'exposant de singularité en un point rend compte du degré local de régularité de la mesure considérée.

Le spectre $f(\alpha)$ des singularités : Le spectre $f(\alpha)$ des singularités associé à la mesure μ est la dimension fractale de l'ensemble des points x_0 tel que $\alpha(x_0) = \alpha$:

$$f(\alpha) = d_f(\{x_0 \in Supp\mu / \alpha(x_0) = \alpha\}) \quad (10)$$

Il décrit la répartition statique des exposants α sur le support de la mesure.

Le spectre $(\alpha, f(\alpha))$ contient un grand nombre d'informations. Les α donnent une mesure locale des singularités, $f(\alpha)$ mesure le comportement fractal des points de même α [51, 52].

L'approche multifractale de détection de contours procède par le calcul direct du spectre des singularités $f(\alpha)$. Elle est particulièrement délicate, dans la mesure où elle nécessite un bon algorithme d'estimation de la dimension fractale, et ce même sur une image comportant peu de points. Les étapes de la détection de contours sont [51, 52, 57, 60] :

- Calculer en chaque point de l'image l'exposant de singularité α , et ceci par l'estimation de la limite du rapport $\frac{\log p_i}{\log \frac{1}{r}}$ par une régression linéaire, où p_i est la mesure d'un sous-ensemble de taille r autour du pixel centré. Ce qui nous permet d'obtenir une image de l'exposant. α est donc calculé comme étant l'estimation de la pente du nuage de points.
- Diviser l'intervalle $[\alpha_{min}, \alpha_{max}]$ en M parties.
- Extraire séparément les pixels correspondants à chaque intervalle.
- Affecter à chaque pixel appartenant à l'intervalle, la valeur 255 et 0 pour les autres. Nous obtenons ainsi M images binaires.
- Calculer la dimension fractale de chaque sous image par la méthode de boîtes [47, 48, 49]. Affecter cette valeur à chaque pixel significatif comme étant $f(\alpha(x,y))$. Nous obtenons alors une image de $f(\alpha)$.
- Opérer une segmentation selon les valeurs de $f(\alpha)$.

La dernière étape s'effectue par l'extraction des pixels dont le $f(\alpha)$ est supérieur à 1 et proche de 2, c.-à-d., $1.4 \leq f(\alpha) \leq 1.8$. On peut extraire les pixels dont le $f(\alpha)$ est proche de 1, ce qui donne une image de contours réguliers. On peut aussi extraire tous les points dont le $f(\alpha)$

est proche de 2 ($1.8 \leq f(\alpha) < 2$), ces points correspondent aux régions homogènes. Mais dans notre cas, on s'intéresse aux points dont $1.4 \leq f(\alpha) \leq 1.8$.

Remarques :

- Pour obtenir une meilleure localisation possible, nous utilisons un voisinage pas plus grand que 7 afin de conserver l'aspect local de α . Dans notre application, nous utilisons un voisinage $V(3)$ c'est à dire 3x3, ainsi le calcul de α portera seulement sur deux valeurs de r à savoir 1 et 3, ce qui nous évite le calcul de la régression ; il en découle un gain en temps de calcul considérable. α est obtenue par la relation suivante :

$$\alpha = \frac{\log \mu(V(3)) - \log \mu(V(1))}{\log 3 - \log 1} \quad (11)$$

- $f(\alpha)$ est obtenue par la méthode de boîtes puisqu'il y a une équivalence avec la dimension de Hausdorff [56].
- Un choix naturel de la mesure μ consisterait à la définir comme la somme des intensités des pixels de la région mesurée. Cette définition respecte l'axiome de la théorie des mesures [56, 60]. Mais nous pouvons utiliser des mesures qui ne respectent pas nécessairement cet axiome. Des exemples de ces mesures pour une région Ω peuvent être donnés par les définitions suivantes :

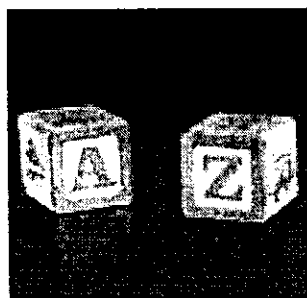
✓ $\mu_{\max}(\Omega) = \max_{i \in \Omega} p(i),$

✓ $\mu_{\min}(\Omega) = \min_{i \in \Omega} p(i),$

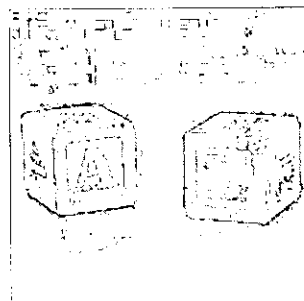
où Ω^* est l'ensemble de Ω où l'intensité est non nulle, $p(i)$ est l'intensité au point i .

Les exposants calculés avec ces mesures donneront des informations différentes quant à la singularité rencontrée.

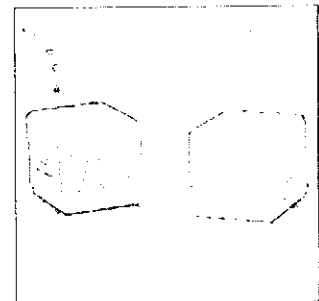
La figure suivante présente les résultats de segmentation multifractale avec les deux mesures *Sum* et *Max*.



(a1)



(b1) Sum



(c1) Max

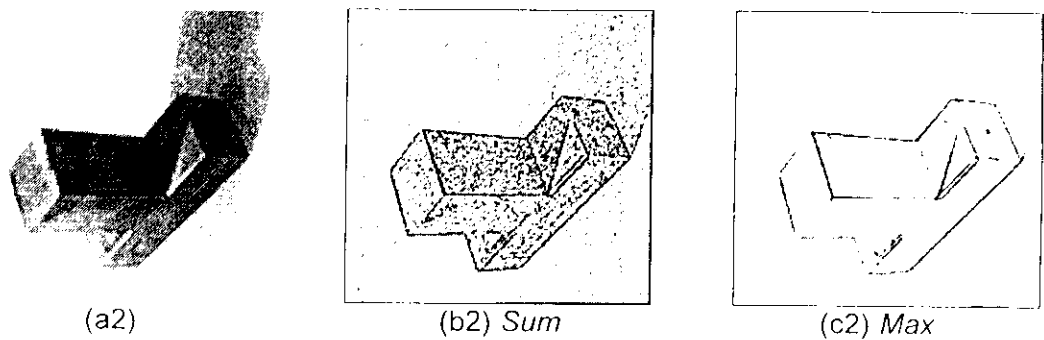


Figure 5 : L'approche multifractale pour la détection de contours.

Nous constatons que les images de contours sont différentes selon le type de la mesure. Ceci est dû au fait que les exposants de singularités calculés avec les différentes mesures donneront des informations différentes quant à la singularité rencontrée.

La mesure *Sum* présente une bonne détection de contours, mais elle peut présenter des points isolés qui ne sont pas considérés comme des contours. Nous pouvons améliorer la détection en faisant suivre la détection par filtrage des points isolés, en utilisant par exemple la dilatation à deux niveaux de gris. La mesure *Max* est moins sensible aux petites variations, et l'apparition de points isolés est moindre [57].

Nous concluons que l'utilisation de l'algorithme multifractal avec les différentes mesures, nous permet d'avoir une variété d'images de contours d'une même image originale, et l'appréciation des résultats est subjective et surtout dépend de ce que nous recherchons.

2.2.3 Détection de contours par la morphologie mathématique

Le principe de base de la morphologie mathématique est de comparer les objets d'une image X à un objet B de référence, de taille et de forme données. Cet objet de référence s'appelle l'élément structurant. Les opérations de base sont la dilatation et l'érosion. La dilatation est habituellement définie en termes de théorie des ensembles comme l'addition de Minkowski de X et de B . L'érosion est l'opération duale, c.-à-d., le complément de la somme de Minkowski du complément de X et de B .

A partir de ces opérations élémentaires, on peut construire des outils plus avancés, tels que l'ouverture et la fermeture. Et l'application itérative de tels filtres permet de définir des opérations plus complexes telles que squelettes, amincissement, épaissement, érosion ultime, ouverture par reconstruction [63, 64, 65, 66, 67, 68, 69] ...

Nous donnons d'abord quelques définitions, ensuite nous présentons des opérations par la morphologie mathématique pour la détection de contours.

2.2.3.1 Morphologie Mathématique à deux niveaux de gris [63, 68]

L'élément de base de toute opération de la morphologie mathématique est l'élément structurant.

Élément structurant : Un élément structurant est une forme de référence qui peut être quelconque et qui est déplacé sur un ensemble de données, c'est généralement une figure géométrique simple dotée d'une origine possédant des caractéristiques particulières.

Il joue le rôle d'une sonde géométrique dont la forme et la dimension permettent d'extraire des formes particulières. Le choix d'un élément structurant n'est pas arbitraire mais découle de plusieurs constatations. En vue de faciliter l'interprétation et la mise en œuvre technologique, les éléments structurants doivent être les plus simples possibles. Souvent, il suffit qu'ils possèdent des propriétés de convexité et d'isotropie, ce choix dépend du problème à résoudre.

Dilatation : Elle consiste à dilater l'image. De ce fait, les points noirs isolés au milieu des parties blanches sont mangés par la dilatation de ces mêmes parties. Ceci est considéré comme une transformation morphologique qui combine des ensembles en utilisant l'addition vectorielle. Donc, si A et B sont deux sous-ensembles de E , alors la dilatation de A par B notée $A \oplus B$ est définie par :

$$A \oplus B = \{c \in E / (a \in A \text{ et } b \in B) c = a + b\} \quad (12)$$

En pratique, ceci est réalisé en faisant passer sur l'image une fenêtre de taille fixe et en effectuant pour chaque pixel de l'image un *OU* logique des pixels formant la fenêtre, à l'exception du pixel traité (pixel central).

Erosion : L'érosion est l'opération duale de la dilatation. Alors dans ce cas, les points blancs d'une image sont mangés par les zones noires qui se trouvent autour. Ceci est donc une transformation morphologique qui combine deux ensembles en utilisant la soustraction vectorielle. Si A et B sont deux sous-ensembles de E , alors l'érosion de A par B notée $A \ominus B$ est définie par :

$$A \ominus B = \{c \in E \setminus c + b \in A \forall b \in B\} \quad (13)$$

Pratiquement, on effectue un *ET* logique entre les pixels contenus dans la fenêtre utilisée, sauf le pixel central.

Notons que les transformations par érosion et dilatation sont des opérateurs « tout ou rien », puisqu'à chaque point x de R^2 correspond une réponse positive ou négative.

Ouverture et Fermeture : Les opérateurs de dilatation et d'érosion peuvent augmenter ou diminuer la taille des objets contenus dans l'image. C'est pourquoi, deux autres opérateurs sont définis qui sont l'ouverture et la fermeture qui représentent une combinaison d'érosion et de dilatation pour le premier, et une combinaison de dilatation et d'érosion pour le second.

La fermeture a pour effet de combler les trous, les chenaux étroits, connecter les objets proches et d'éliminer les pics négatifs qui sont plus étroits que l'élément structurant.

L'ouverture supprime les caps, les petits objets, les pics positifs qui sont plus étroits que l'élément structurant.

2.2.3.2 Morphologie mathématique à plusieurs niveaux de gris

Les opérateurs morphologiques : dilatation, érosion, ouverture et fermeture sont aussi utilisés pour des images à plusieurs niveaux de gris ; pour cela, il faudra remplacer respectivement les notions de *ET* et de *OU* logiques par les notions de maximum et de minimum (*MIN* et *MAX*). Aussi, les définitions des opérateurs d'ouverture et de fermeture d'une image à plusieurs niveaux de gris sont identiques à celles de la morphologie mathématique à deux niveaux de gris.

Notons que le dilaté et l'érodé d'une fonction *f* et *k* sont liés par la relation suivante :

$$\forall x \in R^n, (f \ominus k)(x) \leq f(x) \leq (f \oplus k)(x) \quad (14)$$

Gradient morphologique (ou Gradient symétrisé) : Il donne une approximation du module du vecteur gradient en chaque point. Pour une image *A*, la norme du gradient s'obtient par la transformation :

$$|\nabla I| = \frac{(A \oplus k) - (A \ominus k)}{2} \quad (15)$$

Où *k* est l'élément structurant.

Il suffit donc de dilater et d'éroder l'image en niveaux de gris puisque le gradient s'interprète comme l'étendue, c.-à-d., la différence entre le « max. » et le « min. » sur le disque unité.

Gradient par dilatation : Pour une image *A*, le module du gradient par dilatation avec l'élément structurant *k* s'obtient par la transformation :

$$|\nabla^+ A| = (A \oplus k) - A \quad (16)$$

C.-à-d., la différence entre le dilaté de l'image et l'image elle-même.

Gradient par érosion : Pour une image *A*, le module du gradient par érosion avec l'élément structurant *k* s'obtient par la transformation :

$$|\nabla^- A| = A - (A \ominus k) \quad (17)$$

C.-à-d., la différence entre l'image et son érodé.

Laplacien : Le module du Laplacien $|\Delta A|$ d'une image *A* est la différence entre le module du gradient par dilatation et celui du gradient par érosion :

$$|\Delta A| = |\nabla^+ A| - |\nabla^- A| \quad (18)$$

Jusqu'ici, nous avons défini les opérateurs morphologiques de base, dans ce qui suit, nous présenterons ces différents traitements : Filtrage, détection de contours et amincissement morphologiques.

2.2.3.3 Filtrage morphologique

L'ouverture et la fermeture morphologiques filtrent le bruit quelle que soit son amplitude relative, pourvu qu'elle soit positive pour la première et négative pour la seconde. Ces deux

opérateurs, pris séparément, induisent donc une modification de la moyenne des intensités, d'autant plus importante que la taille de l'élément structurant est grande et que l'amplitude du bruit est forte. Pour pallier cet inconvénient, on peut faire appel au filtrage alterné séquentiel [22] défini, dans le cas discret, de la façon suivante :

Soit $\{O_i\}$ et $\{F_i\}$ une famille d'ouvertures et de fermetures associées à des éléments structurants convexes K_i , de même forme et de taille croissante ($K_i \subset K_{i+1}$).

Le filtre alterné séquentiel est obtenu par :

$$FAS_i = O_i \circ F_i \circ O_{i-1} \circ F_{i-1} \circ \dots \circ O_1 \circ F_1. \quad (19)$$

Ou en permutant les opérations d'ouverture et de fermeture :

$$FAS'_i = F_i \circ O_i \circ F_{i-1} \circ O_{i-1} \circ \dots \circ F_1 \circ O_1. \quad (20)$$

Parmi les filtres morphologiques, le filtre alterné séquentiel est un de ceux qui ont, les meilleures propriétés de réduction de bruit [22].

Notons que toute transformation croissante et idempotente est appelé **filtre morphologique** :

- Une transformation vérifiant : $\forall A, T\{A\} = T$ est dite idempotente.
- Une transformation vérifiant : $A < B \Rightarrow T\{A\} \leq T\{B\}$ est dite croissante.

2.2.3.4 Détection de contours par utilisation du gradient morphologique

L'approche morphologique est adaptée à la prise en compte des critères topologiques pour définir les frontières de régions dans une image. La détection de contours se fait par le calcul de la norme du gradient (ou du laplacien) morphologique [22, 63, 68], ce calcul est très sensible au bruit, il ne peut s'appliquer que sur une image lissée.

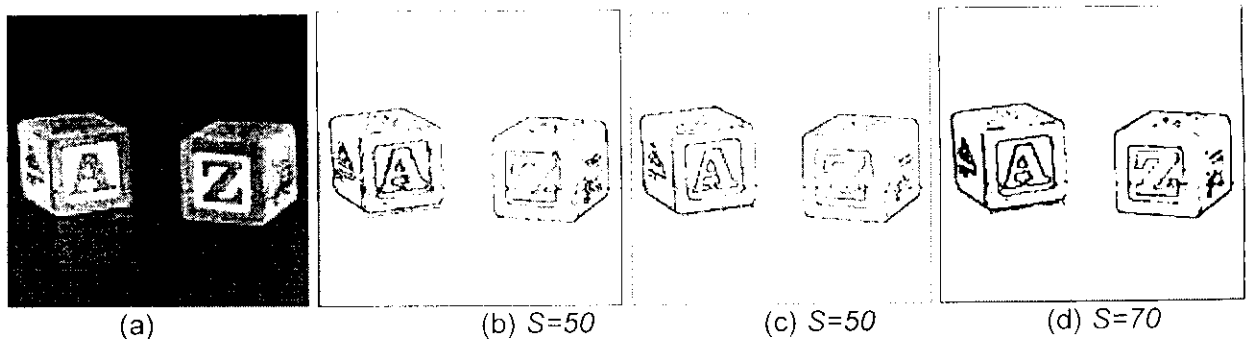


Figure 6 : Détection de contours par les gradients morphologiques, (a) image 2 d'ès, (b) gradient par érosion, (c) par dilatation et (d) symétrique, S : seuil.

Par la figure 6, nous avons une bonne extraction de contours et une sensibilité moindre au bruit grâce au filtrage par le filtre alterné séquentiel. Notons que les contours obtenus par le gradient morphologique symétrique sont épais.

En effet, les contours obtenus après un simple seuillage peuvent être épais, il faut donc les amincir, par exemple par un amincissement morphologique homo-topique.

Où « \setminus » représente la différence ensembliste.

Tout pixel ayant une configuration donnée (définie par l'élément structurant) est éliminé de X . Notons qu'il existe un très grand nombre d'éléments structurants, les plus utilisés sont répertoriés dans l'alphabet de Golay [68]. Voici ci dessous les éléments structurants L et E ; les pixels appartenant à S^0 sont notés 0 , ceux appartenant à S^1 sont notés 1 , les $*$ sont des pixels qui ne sont pas pris en compte.

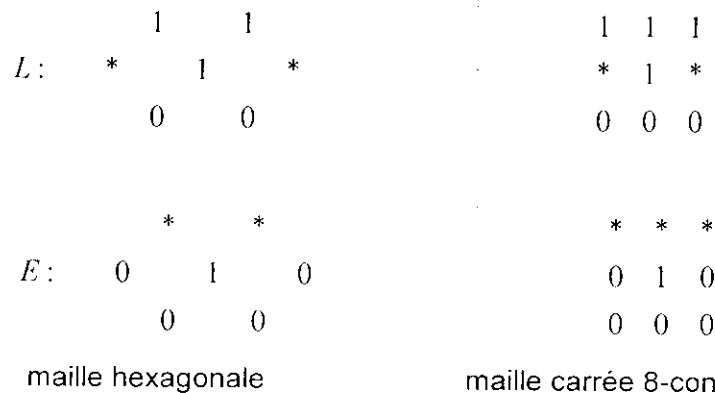


Figure 7 : Les éléments structurants L et E .

L'élément structurant L est homotopique, c.-à-d., il préserve le nombre de composantes connexes et le nombre de « trous » de chaque composante connexe, alors que E ne l'est pas.

Les éléments structurants n'étant en général pas isotrope, on itère des amincissements avec les 5 éléments structurants déduits de ceux de l'alphabet de Golay par rotation de $\pi/6$ en maille hexagonale ou les 7 éléments déduits par rotation $\pi/4$ en maille carrée.

En effet, l'amincissement effectué avec l'élément structurant L (et ses rotations) en maille carrée itéré jusqu'à stabilisation consiste à remplacer le niveau de gris $I[i, j]$ par la valeur I_0 si $I_0 < I[i, j] \leq I_1$, sachant que :

$$I_0 = \max\{I[i-1, j-1], I[i-1, j], I[i-1, j+1]\} \quad (24)$$

$$I_1 = \max\{I[i+1, j-1], I[i+1, j], I[i+1, j+1]\} \quad (25)$$

⁴ Le translaté de X par t est donné par : $X_t = \{x+t, x \in X\}$

toutes les directions (θ en maille carrée). Ce qui permet d'amincir les contours sur toutes les directions, et les contours obtenus sont beaucoup mieux amincis qu'avec la méthode de suppression des points non maximums⁵, puisque cette dernière traite uniquement les points sur la direction du gradient. Notons aussi que l'utilisation de l'élément structurant L est moins sensible au bruit, et le suivi d'ébarbulage n'est souvent pas nécessaire.

2.2.3.6 La ligne de partage des eaux

Si l'on considère l'image comme une surface dans l'espace de dimension 3, on peut utiliser la terminologie de la géographie qui définit la ligne de partage des eaux comme la crête qui forme la limite entre deux bassins versants. Celle-ci s'obtient par amincissement morphologique effectué avec l'élément structurant L (et ses rotations) itéré jusqu'à stabilisation.

En effet, en itérant jusqu'à stabilisation, les lignes de crêtes gardent leurs niveaux de gris d'origine et entourent des régions dont le niveau de gris est uniforme et égal au niveau de gris le plus bas de la région. Un simple seuillage suffit alors à extraire la ligne de partage des eaux. D'autres algorithmes existent et permettent la construction de la ligne de partage des eaux sont donnés dans [66, 67, 69, 70].

Sur une image non texturée comportant des objets sombres sur un fond clair (ou vice-versa), la ligne de partage des eaux fournit un partitionnement en zones d'influences des objets, même si la frontière entre les objets est très faiblement marquée.

Ce traitement est particulièrement adapté pour séparer des objets très proches, pour lesquels la frontière ne se manifeste que par quelques niveaux de gris de différence par rapport aux objets. Pour séparer des objets pour lesquels la frontière est légèrement plus foncée que les objets eux-mêmes, on opérera une inversion préalable des niveaux de gris de l'image.

La figure 8 présente la ligne de partage des eaux de deux images de scènes réelles :

⁵ Elle consiste en la recherche des points adjacents qui se trouvent dans la direction du gradient. Et si le gradient au point (x, y) est supérieur à la fois à ces points adjacents, alors il est conservé, sinon il est supprimé (il n'est pas un point contour)

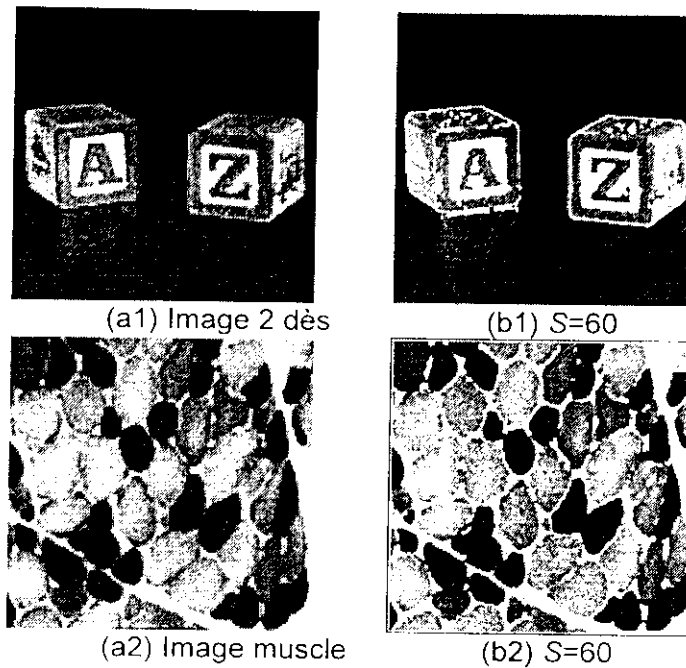


Figure 8 : La ligne de partage des eaux.

Notons que la ligne de partage des eaux LPE fournit souvent des contours fermés. Ce fait est de première importance puisque nous ne devons pas nous inquiéter du problème de fermeture des contours des objets détectés, ce qui pourrait se produire en utilisant des méthodes de détection classiques.

La LPE est une méthode générale qui peut être appliquée à beaucoup de situations. Nous donnons un exemple d'application pour la séparation des objets qui se touchent, mais en fait, cet exemple est un petit choix des domaines dans l'analyse d'image où cette technique a été employée efficacement.

2.2.4 Autres approches de détection de contours

Il existe bien d'autres méthodes de détection de contours, nous pouvons citer :

L'approche variationnelle ou contours actifs [71, 72, 73], les contours sont modélisés par des courbes élastiques qu'il s'agit d'ajuster le mieux possible. Pour cela on associe une énergie dont la définition est étroitement liée à celle des points contours. Les minima locaux de cette énergie correspondent aux contours s'ajustant le mieux au modèle utilisé.

L'inconvénient majeure de cette approche réside dans la phase d'initialisation manuelle de la courbe de départ. Elle présente également une sensibilité au choix de paramètres de rigidité et d'élasticité.

Dans [74], une méthode est proposée par minimisation d'une fonction de coût pour détecter les contours en utilisant le recuit simulé. L'algorithme se décompose en deux étapes:

Tout d'abord, une matrice de contours potentiels est créée. Douze noyaux de convolution sont utilisés, chacun d'eux étant associé à un profil de contours et réalise un filtrage suivi d'une suppression des non-maxima locaux.

Puis une fonction d'énergie est définie comme une somme pondérée de facteurs de coût.

D'autres méthodes de détection de contours existent qui utilisent les modèles markoviens, la programmation dynamique, les réseaux de neurones [22, 75]...

2.2.5 Conclusion

Nous avons présenté et étudié d'une manière approfondie des opérateurs de détection de contours basés sur différents principes. Nous énonçons que :

- ✓ L'utilisation de la dimension fractale locale permet la détection des brusques transitions présentes dans l'image, cependant, les vrais contours, c.-à-d., ceux correspondant aux limites des objets, ne sont pas toujours détectés. Et nous constatons souvent des contours interrompus, ceci est dû à la nature même de l'algorithme qui procède par le calcul local à l'échelle du pixel de la dimension fractale.
- ✓ Avec l'algorithme multifractal, différentes images de contours sont obtenues selon les mesures utilisées, ce qui permet d'avoir une variété d'images de contours d'une même image originale, et l'appréciation des résultats est subjective, et surtout dépend de ce que nous cherchons. Ces algorithmes multifractals présentent l'avantage de la non nécessité de paramètres d'entrée, tel que : seuils, taille des masques, ...
- ✓ Avec les gradients morphologiques, une bonne détection des contours est obtenue, ils sont souvent continus et fermés. Notons particulièrement, les contours fermés et amincis obtenus par la ligne de partage des eaux.

Notons aussi, qu'avec l'algorithme multifractal, il est possible de détecter les contours dans des zones texturées, sachant que les exposants de singularité α mesurent la texture [51, 52]. Cependant, les autres approches déjà présentées sont peu adaptées aux images texturées pour lesquelles il est préférable d'exploiter des approches régions [39, 40, 41, 42].

2.3 Approches de segmentation en régions homogènes

2.3.1 Techniques de seuillage

Le seuillage est considéré comme étant la plus simple technique de classification. Seuiller une image équivaut à regrouper les niveaux de gris en classes selon un critère de similitude. La difficulté de cette approche consiste à déterminer le nombre de classes présentes dans l'image.

Puisque les seuils de classification sont positionnés sur les minimums locaux de l'histogramme. Et l'identification peut s'effectuer par analyse de l'histogramme, c'est une méthode simple, rapide et bien adaptée aux images peu bruitées. Mais le bruit introduit sur l'histogramme de nombreuses irrégularités pouvant mettre en défaut la détection des seuils.

On peut se tourner vers d'autres méthodes qui maximisent ou minimisent une fonction traduisant un critère et qui est définie par rapport à la quantité de pixels de chaque niveau de gris, qui peuvent être placés dans les classes suivant lesquelles l'image a été partagée. Ce sont alors des critères basés sur l'histogramme des niveaux de gris de l'image. Les algorithmes de multiseuillage les plus représentatifs sont ceux de Fisher et de Bhattacharya [22] qui sont fondés sur le comportement du logarithme de la fonction histogramme de l'image.

Plusieurs méthodes sont présentées dans [4, 22, 24, 76, 77, 78], nous présenterons par la suite quelques unes.

2.3.1.1 Maximisation de l'entropie a Posteriori

La notion d'entropie est utilisée en analyse d'images pour segmenter des images de manière à maximiser la qualité de l'information retenue.

Soit I une image composée de M pixels, représentés sur N niveaux de gris. Soit f_i la fréquence de chacun des niveaux de gris et $p_i = f_i / M$ la probabilité associée à chacun de ces niveaux.

La formule suivante nous fournit l'entropie a priori de I [24]:

$$H(p_0, p_1, \dots, p_{N-1}) = - \sum_{i=0}^{N-1} p_i \ln p_i \quad (26)$$

Après la binarisation de la formule (26), l'image sera partagée en deux classes, notées I_1 et I_2 par un seuil s . La probabilité associée à chacune de ces classes est :

$$p_1 = \sum_{i=0}^s f_i / M \quad \text{et} \quad p_2 = \sum_{i=s+1}^{M-1} f_i / M \quad (27)$$

L'entropie a posteriori de I binarisée sera donc :

$$H(I_1, I_2) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \quad (28)$$

En maximisant $H(I_1, I_2)$, on maximise l'information a posteriori contenue dans I binarisée. Cette opération fournit la valeur de s , comprise entre 0 et $N-1$, qui partage l'image I de manière à optimiser l'information retenue, c.-à-d., séparer le mieux possible les pixels clairs des pixels foncés. Le seuil s retenu sera alors celui qui approche le plus possible p_1 de p_2 , dans le cas où l'égalité est possible, on a :

$$\sum_{i=0}^s p_i = \sum_{i=s+1}^{N-1} p_i \quad (29)$$

et par conséquent $H(I_1, I_2) = -2 \cdot 0,5 \cdot \log_2 0,5 = \log_2 2 = 1$, qui est sa valeur maximale, c.-à-d., l'entropie d'une expérience ayant deux évènements équiprobables.

2.3.1.2 Algorithme de binarisation automatique

Le problème de la binarisation est le choix du seuil, ce dernier doit diviser l'image en deux classes : Objet et fond. Sachant qu'à partir d'une image en niveau de gris, nous pouvons obtenir plusieurs images binaires, il est alors difficile de dire laquelle est correcte. Dans des cas simples, la moyenne ou la valeur médiane des pixels de l'image peut jouer le rôle d'un seuil. Aussi, si l'histogramme est bimodal, nous choisissons alors le seuil comme étant le point le plus inférieur entre les deux pics.

Nous présentons un algorithme qui considère initialement la moyenne de l'image comme un seuil. Et le seuil final est déterminé par une procédure itérative. Les différentes étapes de l'algorithme sont les suivantes :

1. Créer l'histogramme $H(i)$, où $H(i)$ correspond au nombre de pixels ayant le niveau de gris i .
2. Calculer la moyenne de l'image et la considérer comme seuil initial :

$$s = \frac{1}{L.M} \sum_{i=0}^{L-1} \sum_{j=0}^{M-1} I(i,j) \quad (30)$$

M étant la hauteur de l'image I , L sa largeur.

3. Calculer $T_{objet} = t_1 / 2.t_2$

$$t_1 = \sum_{i=0}^S i * H[i] \quad t_2 = \sum_{i=0}^S H[i] \quad (31)$$

4. Calculer $T_{fond} = t'_1 / 2.t'_2$

$$t'_1 = \sum_{i=S+1}^{255} i * H[i] \quad t'_2 = \sum_{i=S+1}^{255} H[i] \quad (32)$$

5. Si le seuil ' s ' est égal à $(T_{objet} + T_{fond})$ alors ' s ' est le seuil recherché. Dans le cas contraire, affecter au seuil la valeur $(T_{objet} + T_{fond})$ et recommencer l'itération à partir de la troisième étape.

2.3.1.3 Algorithme multiseuillage de Fisher

La méthode de Fisher consiste à localiser les seuils comme les séparateurs des distributions ; ceci est réalisé à l'aide d'un critère de minimisation de la somme des inerties associées aux différentes classes. La méthode de Fisher nécessite de connaître *a priori* le nombre de classes à détecter [22].

La méthode de Fisher détermine la partition des N niveaux de gris en k classes qui minimise la somme des variances des niveaux de gris de chacune de ces classes, donnée par :

$$\min_P \left\{ \sum_{l=1}^k \sum_{i \in P_l} [f_i (i - G(P_l))^2] \right\} \quad (33)$$

Où $P = \{P_1, P_2, \dots, P_k\}$ est une partition des niveaux de gris $0, 1, 2, \dots, N-1$, f_i est le nombre de points de la partition qui possèdent le niveau de gris i et $G(P_l)$ est la moyenne pondérée des valeurs de gris de la classe l , donnée par :

$$G(P_l) = \frac{\sum_{i \in P_l} f_i \cdot i}{\sum_{i \in P_l} f_i} \quad (34)$$

L'évaluation est faite sur toute la dynamique des niveaux de gris, et le nombre de classes est un paramètre d'entrée.

Notons qu'ils existent plusieurs autres méthodes de recherche de seuils, tel que: les méthodes basées sur la théorie de l'information (la méthode de Pun [76], la méthode de Kapur [77], ...), la méthode préservant les moments [43], ...

Ces méthodes de seuillage donnent toutes des résultats satisfaisants suivant la classe d'images pour laquelle elles ont été développées. Cependant, elles ne sont pas toutes automatiques ; certaines sont interactives et nécessitent un opérateur extérieur pour l'obtention d'un seuil optimal. L'automatisation d'une technique de seuillage présente encore des difficultés car un mauvais seuillage entraîne systématiquement une extraction erronée des paramètres caractérisant chaque région (exemple : mauvais tracé du contour).

Notons que l'application des différents algorithmes sur une même image ne donne pas nécessairement les mêmes valeurs des seuils. Par exemple, pour l'image des 2 dés, nous trouvons $S=104$ avec l'algorithme automatique et $S=129$ avec l'algorithme de Fisher. Donc, il

existe N points qui sont différemment classés par les deux segmentations, avec $N = \sum_{i=S+1}^{S+1} H[i]$ et H est l'histogramme de l'image.

Par la figure suivante, nous présentons les résultats de seuillage.

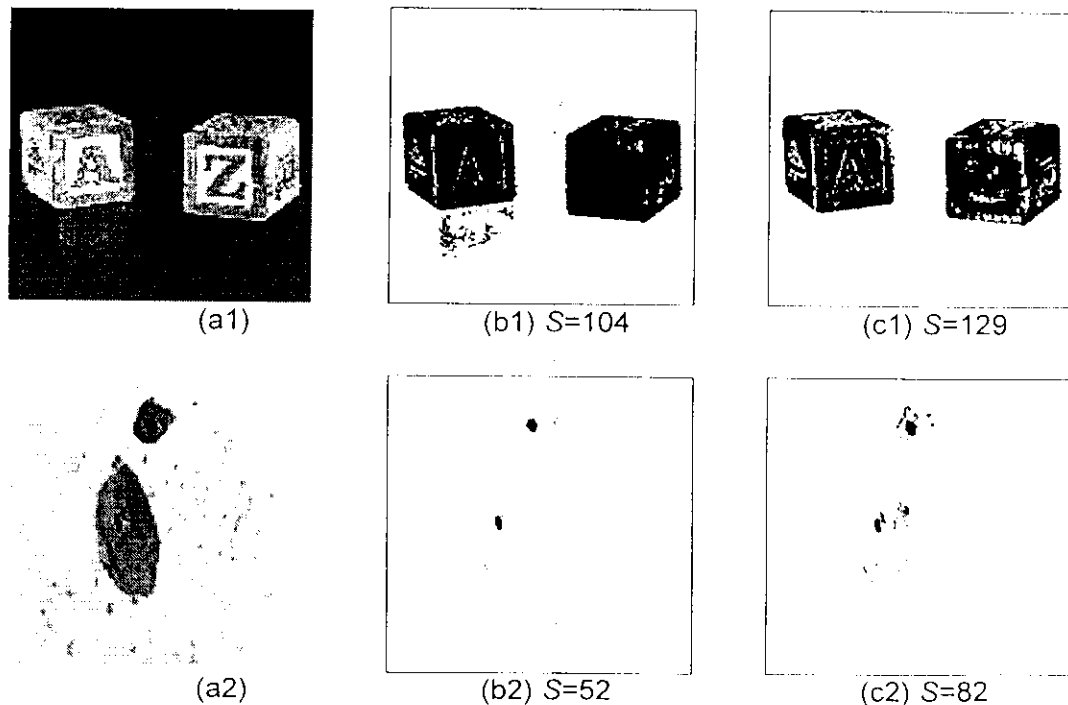


Figure 9 : Segmentation par seuillage des images (ai), avec (bi) l'algorithme automatique et (ci) l'algorithme de Fisher.

Avec l'image 2 dés, nous obtenons de meilleurs résultats avec le seuil estimé par la méthode de Fisher, en effet, nous constatons une nette séparation des objets de la scène puisqu'il y a moins de points de confusion entre les deux principales classes de l'image : Objets et fond. Cependant, avec l'image de cellules, l'image des noyaux obtenue avec le seuil estimé

par la méthode de seuillage automatique est meilleure, pour la même raison que précédemment.

Nous concluons que les deux méthodes peuvent présenter des seuils différents, et elles peuvent être exploitées ensemble pour recherche le meilleur seuil.

Dans ce qui suit, nous exploitons la logique floue pour améliorer le seuillage en utilisant les degrés d'appartenance de chaque intensité à chacune des classes, et plus particulièrement par analyse des points dont les intensités sont au voisinage des seuils.

2.3.2 Segmentation par la logique floue

Dans les techniques de seuillage, un point ne peut appartenir à la fois à un ensemble et à son complémentaire ; il ne peut pas, non plus, n'appartenir à aucun des deux. Les ensembles flous violent à un certain niveau ces lois du tiers exclu et de non-contradiction [24, 79, 80]. En effet, contrairement à la logique classique, la logique floue permet de manipuler d'autres valeurs de vérité que le « vrai » et le « faux » absolus. Les éléments possèdent des degrés d'appartenance à des ensembles ou des classes. La seule contrainte est que la somme des degrés d'appartenance d'un élément à des ensembles complémentaires doit être égale à l'unité.

Notons que la fonction d'appartenance n'est pas de nature aléatoire. Il ne faut pas confondre ses valeurs avec des pourcentages issus des probabilités car ces dernières mesurent si quelque chose risque de se produire alors que le flou mesure le degré d'existence d'un fait ou d'une condition.

Nous allons introduire des mécanismes de la logique floue en segmentation d'images avec l'objectif d'obtenir des résultats précis à travers deux stratégies de segmentation : le seuillage flou et un algorithme de classification fuzzy C-mean. Mais avant, nous présentons des définitions de base nécessaires pour la compréhension de ces algorithmes.

2.3.2.1 Segmentation par seuillage floue

Au moment de la séparation des niveaux de gris en deux classes par un seuil s , il est possible de prendre en compte l'incertitude d'appartenance de quelques pixels aux deux classes, principalement ceux qui ont un niveau de gris très proche de s [24, 79, 80].

La logique floue permet de laisser les niveaux de gris appartenant à l'intervalle $[s-k, s+k]$ attachés aux deux classes de manière floue. On supposerait alors qu'en dehors de cet intervalle les niveaux de gris n'ont aucune incertitude quant à la classe à laquelle ils appartiennent. Alors, deux nouveaux problèmes surgissent :

- ✓ Comment choisir un tel intervalle ?
- ✓ Comment définir les degrés d'appartenance des niveaux de gris de cet intervalle aux deux classes ?

En fait, si l'intervalle $[s-k, s+k]$ est trop petit, il est possible que des seuils non représentatifs soient détectés, par contre si cet intervalle est trop grand, des bons seuils

peuvent être éliminés. Pour éviter cela, il est important que k soit inférieur, en restant proche, de la moitié de la distance entre deux maxima de l'histogramme [24]. L'hypothèse de la convexité de l'histogramme entre les maxima a été utilisée. Pour les histogrammes ayant plus de deux maxima ou alors n'en ayant qu'un seul, cette règle ne donne pas toujours de bons résultats. En tout cas, pour éviter la détection des maxima locaux, il est conseillé de lisser l'histogramme d'abord.

Le deuxième problème peut facilement être contourné par l'utilisation de la fonction S_z (connue comme fonction S de Zadeh). Cette fonction est définie de manière que, pour tout élément x dans un intervalle $[a, c]$ inclut dans $[0, N-1]$, la valeur de $S_z(x)$ soit dans $[0, 1]$:

$$S_z(x) = \begin{cases} 0 & \text{si } x \leq a \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{si } a \leq x \leq b \\ 1-2\left(\frac{x-c}{c-a}\right)^2 & \text{si } b \leq x \leq c \\ 1 & \text{si } x \geq c \end{cases} \quad (35)$$

Le point $b=(c+a)/2$, pour lequel S_z possède la valeur 0.5, est appelé point de croisement et la valeur de $(c-a)$ est dite « largeur de bande ».

Soit s le point de croisement et $2k$ la largeur de bande, la fonction S_z peut alors être définie comme suit (voir figure 10):

$$S_z(x) = \begin{cases} 0 & \text{si } x \leq s-k \\ \frac{(x-(s-k))^2}{2k^2} & \text{si } s-k \leq x \leq s \\ 1-\frac{1}{2k^2}(x-(s+k))^2 & \text{si } s \leq x \leq s+k \\ 1 & \text{si } x \geq s+k \end{cases} \quad (36)$$

Cette fonction possède la particularité de ne pas avoir une grande variation de ses valeurs dans un petit intervalle, en dépit d'une petite variation en dehors de cet intervalle. Il est suggéré dans [24] que deux conditions soient satisfaites :

- ✓ Le respect de la symétrie par rapport à s :

$$\mu(s-x) + \mu(s+x) = 1 \quad \forall x \leq k \quad (37)$$

- ✓ Le respect de $h_{inf}(x) \leq \mu(x+s) \leq h_{sup}(x) \quad \forall x \leq k$ où les fonctions-limites h_{inf} et h_{sup} sont données par :

$$h_{inf}(x) = \begin{cases} 0 & \text{si } \min \leq x \leq \min + \varepsilon \\ x - \varepsilon & \text{si } \min + \varepsilon \leq x \leq \max \end{cases} \quad (38)$$

$$h_{sup}(x) = \begin{cases} x + \epsilon & \text{si } \min \leq x \leq \max - \epsilon \\ 1 & \text{si } \max - \epsilon \leq x \leq \max \end{cases} \quad (39)$$

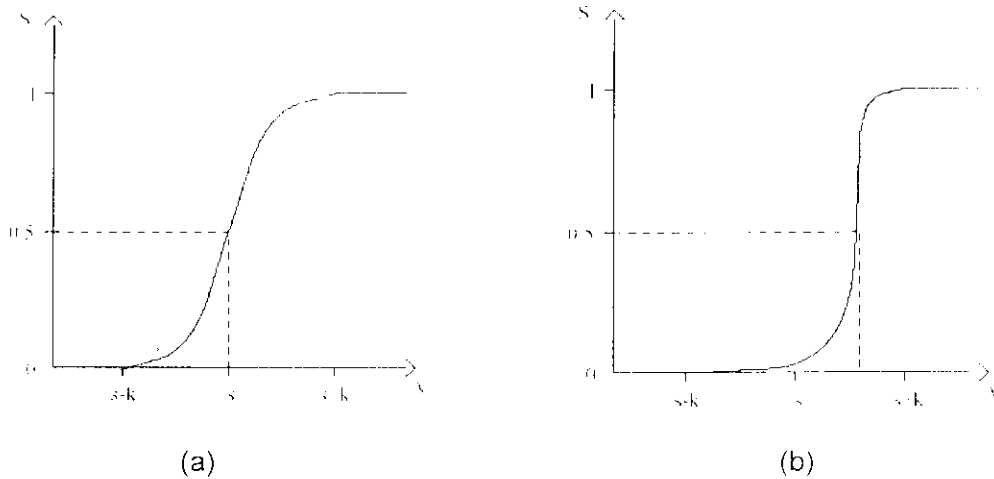


Figure 10 : (a) Fonction S_z , (b) fonction d'appartenance non symétrique.

La figure suivante montre les fonctions-limites h_{inf} et h_{sup} développées pour $\min=s-k$, $\max=s+k$ et $\epsilon=0.25$.

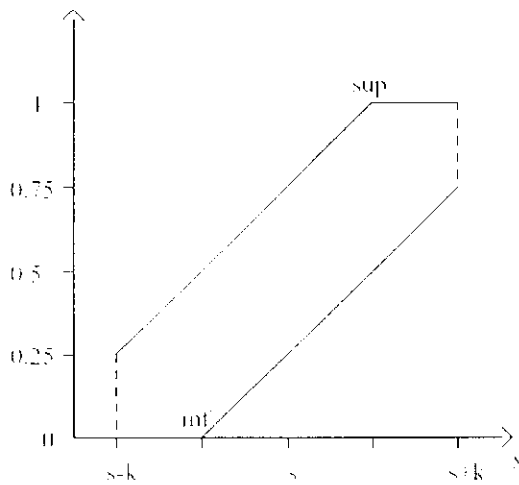


Figure 11 : Fonctions limites

Remarquons que la fonction S_z satisfait ces conditions. Nous aurons ainsi pour chaque seuil s le degré d'appartenance de tous les niveaux de gris aux deux classes : le fond et l'objet, comme suit :

$$\mu(\text{fond}) = \begin{cases} 0 & \text{si } x < s-k \\ S_z(s,x) & \text{si } s-k \leq x \leq s+k \\ 1 & \text{si } x > s+k \end{cases} \quad (40)$$

$$\mu(\text{objet}) = \begin{cases} 1 & \text{si } x < s-k \\ 1-S_z(s,x) & \text{si } s-k \leq x \leq s+k \\ 0 & \text{si } x > s+k \end{cases} \quad (41)$$

Le degré d'appartenance de chaque niveau de gris présent dans une image étant donné par la fonction S_z (36). A chaque valeur de croisement possible correspond une partition floue de l'image.

Soit une image X ayant un histogramme des niveaux de gris bimodal, le point de croisement s choisi comme seuil optimal séparant l'image en deux classes est celui pour lequel l'ambiguïté est maximale, c.-à-d., $S_z(x) = 0.5$. Les différents niveaux de gris seront séparés en deux classes :

$$\begin{cases} x \in [0, s-1] \text{ où } S(x) < 0.5 \\ x \in [s+1, N-1] \text{ où } S(x) > 0.5 \end{cases} \quad (42)$$

Notons que nous pouvons utiliser d'autres formes de fonctions d'appartenances (Triangulaire, trapézoïdale, ...) que la fonction S [79]. Des études ont montré que le choix du type de la fonction d'appartenance n'influe pas sur la qualité du résultat.

Dans ce qui suit, nous présentons un algorithme récursive de segmentation par la logique floue.

2.3.2.2 Algorithme Fuzzy C-Means

Le fuzzy c-means est un algorithme itératif, non supervisé de classification floue. Il est basé sur un critère quadratique qui sert à classifier des formes en classes non-disjointes en permettant de conserver longtemps un volume d'information important, sans avoir à prendre la décision prématurément.

Le critère quadratique à minimiser est la somme pondérée, pour toutes les classes formées, des écarts quadratiques intra-classes [24, 79, 80].

Soit $X = \{x_1, x_2, \dots, x_M\}$ un ensemble fini de formes de dimension p , c un entier appartenant à $\{2, 3, \dots, M\}$ représentant le nombre de classes et $U = \mu_{ik}$ une partition floue de X en c classes, où chaque μ_{ik} est le degré d'appartenance $\mu_k(x_i)$ de la forme x_i à la classe k .

Supposons $\sum_{k=1}^c \mu_{ik} = 1 \quad \forall i$, alors le critère quadratique de classification J_m est défini par :

$$J_m(U, V) = \sum_{i=1}^M \sum_{k=1}^c (\mu_{ik})^m (d_{ik})^2 \quad (43)$$

où m est le facteur de flou ($1 \leq m < \infty$), d_{ik} est une distance quelconque entre la forme x_i et le vecteur $v = \{v_1, v_2, \dots, v_p\}$ des centres des classes.

Lorsque m tend vers 1 et d est la distance euclidienne, l'algorithme FCM est connu comme Hard C-Fuzzy (HCF). Dans ce cas, il fournit la partition binaire optimale.

Algorithme FCM :

1. Initialisation

- Fixer les paramètres c ($2 < c \leq M$), $m \in [1, \infty[$ et ϵ (seuils de convergence).
- Initialiser la partition floue U ou le vecteur v des centres des classes.

2. Répéter jusqu'à $|J_m^k - J_m^{k-1}| \leq \epsilon$

- Calculer les centres v_k des c classes :

$$v_k = \frac{\sum_{i=1}^M (\mu_{ik})^m x_i}{\sum_{i=1}^M (\mu_{ik})^m}$$

- Calculer la partition U floue :

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}}$$

Effet des paramètres de l'algorithme : Notons qu'il n'y a pas de règles pour fixer la valeur de m car il n'existe pas de base théorique pour l'optimisation de ce paramètre. Cela permet de mettre en valeur l'ambiguïté existante dans l'ensemble à classer, ou au contraire, de l'atténuer. Le facteur de flou m interfère sur deux caractéristiques de l'algorithme : la rapidité de convergence décroît avec l'augmentation de m , en même temps que l'apport de chaque élément dans le calcul des centres des classes décroît.

Pour les valeurs de m supérieures à 2, les partitions tendent, lorsque m croît, vers le centre de gravité de l'espace des partitions floues, c.-à-d., vers la partition $U = \left[\frac{1}{c} \right]$, ce qui n'offre pas de réel intérêt. Pour les valeurs de m proches de 1, les degrés d'appartenance sont très stricts (proches des valeurs binaires) et ne traduisent pas l'ambiguïté d'affectation des formes de X . Enfin, les valeurs de m prises dans l'intervalle $[1.5, 2]$ permettent d'obtenir des résultats intéressants et d'interprétation aisée.

Le nombre d'itérations de l'algorithme FCM augmente avec la précision demandée sur les valeurs des centres (ou de degrés d'appartenance, si le test d'arrêt est fait en fonction de celles-ci). Pour les formes entières (les niveaux de gris, par exemple), un seuil de convergence de 0.01 donne des erreurs inférieures à 0.5%, ce qui nous apporte un bon compromis entre la rapidité de convergence et la précision des résultats.

En classification floue, l'appartenance d'un élément à une classe est d'autant plus forte que le degré d'appartenance à cette classe est proche de 1 et que ses degrés d'appartenance aux autres classes sont proches de 0. La meilleure classification floue réalisable est donc la classification qui correspond le plus à une partition binaire.

Notons aussi que des phénomènes de scission ou d'agrégation des classes sont observables au fur et à mesure que l'on exécute l'algorithme FCM pour 2, 3, ..., c classes.

La convergence de l'algorithme FCM vers un minimum local est assurée quelque soit la configuration initiale choisie, à condition que plusieurs centres ne soient pas initialisés aux mêmes valeurs.

Algorithme FCM appliqué à la segmentation d'images : Cet algorithme de segmentation par classification floue de pixels est constitué de deux étapes :

1. La division de l'ensemble des pixels en un nombre donné de classes non-disjointes. Durant cette première étape, on cherche à conserver le plus longtemps possible les informations utiles, jusqu'au moment de la prise de décision (défuzzification).
2. La définition des différentes régions contenues dans l'image, qui pourra s'effectuer par un seuillage par rapport à un degré d'appartenance donné ou plus simplement en choisissant d'attribuer chaque pixel à la classe pour laquelle son degré d'appartenance est maximal.

Comme dans les techniques de segmentation par classification, il n'est pas nécessaire de classifier tous les pixels d'une image, mais de classer simplement les différentes valeurs de niveaux de gris que l'on trouve dans celle-ci, l'algorithme FCM peut être adapté au cas de la classification floue des N niveaux de gris d'une image. Cet algorithme permet une réduction importante du nombre d'informations manipulées, et cela parce que si l'image n'est pas étalée sur toute la dynamique disponible, on ne considère que les niveaux de gris présents dans l'image. De plus, le temps de traitement d'une image 512x512 est sensiblement le même que pour une image 256x256 ou 128x128 si elles sont quantifiées sur le même nombre de niveaux de gris, non nécessairement les mêmes.

Algorithme FCM adapté :

1. Initialisation
 - Fixer les paramètres c ($2 \leq c \leq M$), $m \in [1, \infty[$ et ε (seuils de convergence).
 - Initialiser la partition floue U ou le vecteur v des centres des classes.
 - Calculer la fréquence f_i de chaque niveau de gris dans l'image.
2. Répéter jusqu'à $|J_m^m - J_m^{m-1}| \leq \varepsilon$
 - Calculer les distances d_{ik} entre tous les niveaux de gris i ($i=1, 2, \dots, N-1$) et les centres v_k des classes ($k=1, 2, \dots, c$)
 - Calculer la partition U floue :

$$\mu_{ik} = \frac{1}{\sum_{l=1}^c \left(\frac{d_{il}}{d_{ik}} \right)^{\frac{1}{m-1}}}$$
 - Calculer les centres v_k des c classes :

$$v_k = \frac{\sum_{i=0}^{N-1} (\mu_{ik})^m \cdot f_i}{\sum_{i=0}^{N-1} (\mu_{ik})^m \cdot f_i}$$

Par la figure suivante, nous donnons des résultats d'application de l'algorithme FCM adapté.

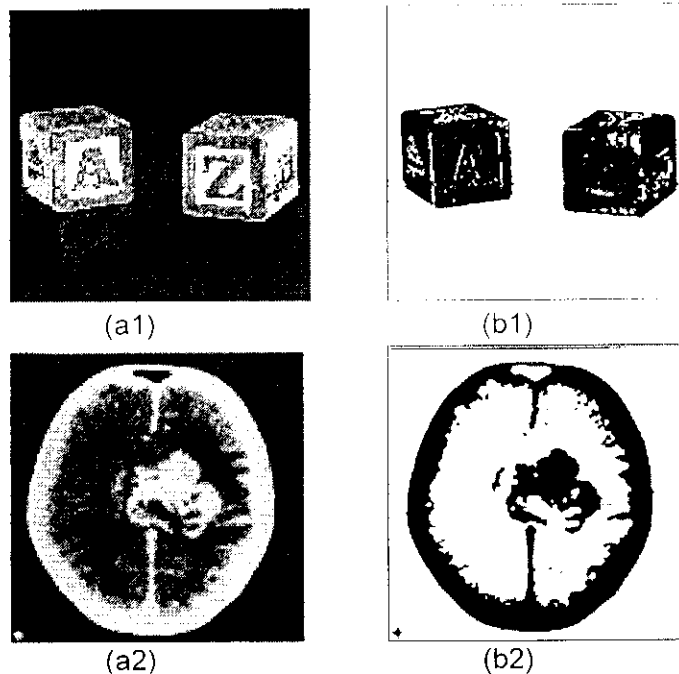


Figure 12 : Segmentation par l'algorithme Fuzzy C-Means.

D'après ces résultats, une bonne segmentation est obtenue et il y a peu de confusions entre les classes de l'image.

Dans ce qui suit, nous présentons une méthode de segmentation qui ne considère pas uniquement l'intensité du pixel mais elle tient aussi compte de la corrélation entre pixels voisins.

2.3.3 Algorithme de relaxation

Pour un ensemble d'objets contenus dans une scène dont l'identification est ambiguë, il est toujours possible d'utiliser des relations sur ces objets pour réduire ou éliminer cette ambiguïté. Dans les méthodes de seuillage, l'affectation d'un pixel à une des classes ne tient compte que du niveau de gris de celui-ci. Cette approche pourrait se justifier si le niveau de gris d'un pixel était indépendant de ceux de ses voisins. Une image étant formée d'entités cohérentes (c.-à-d., le contenu d'une image n'est pas un phénomène aléatoire non corrélé), cette indépendance n'est pas vérifiée [37, 38]. Les méthodes contextuelles se proposent de tenir compte de cette corrélation entre pixels voisins. La méthode contextuelle la plus célèbre est sans aucun doute la relaxation probabiliste, son principe est le suivant :

On détermine tout d'abord un étiquetage initial sous la forme d'un vecteur de probabilité en chaque pixel qui traduit l'appartenance potentielle à chacune des classes. Cet étiquetage est ensuite modifié itérativement en tenant compte de l'étiquetage des voisins d'un point. Pour cela,

une notion de compatibilité entre étiquetages sur des sites voisins est utilisée. Dans ce qui suit, nous présentons en détail l'algorithme.

ALGORITHME :

Soit un étiquetage initial $P_{i,0}$ pour chaque pixel i de l'image [33, 83, 84, 85, 86]. $P_{i,0}$ est le vecteur de probabilités tel que $P_{i,0}(z)$ représente la probabilité d'appartenance du point i à la classe C_z pour z variant de 1 à c , où c est le nombre de classes fixé *a priori*.

L'utilisation de probabilité impose la contrainte :

$$\sum_{z=1}^c P_{i,k}(z) = 1 \tag{44}$$

quelque soit l'itération k .

L'étiquetage initial : Le choix de l'étiquetage initial est apparu depuis la naissance de cette méthode. Convergence, rapidité, qualité des coefficients de compatibilité dépendent de lui. C'est pourquoi on utilise souvent la relaxation pour parfaire un étiquetage initial obtenu par une méthode classique de seuillage. A partir d'un ensemble de seuils, on définit N classes sur lesquelles on peut calculer la moyenne μ_k et l'écart type σ_k . La probabilité initiale pour que le pixel i appartienne à la classe C_z est obtenue par :

$$P_{i,0}(z) = \frac{1/d_{iz}}{\sum_z^c 1/d_{iz}} \tag{45}$$

où d_{iz} est la distance, au sens de Mahalanobis, entre le pixel i et la classe C_z :

$$d_{iz} = \frac{[\mu_z - I(i)]^2}{\sigma_z^2} \tag{46}$$

Nous pouvons exploiter les techniques de seuillages présentés précédemment afin d'assurer un meilleur étiquetage initial aux pixels de l'image [87]. Sachant que la qualité de segmentation dépend étroitement de cet étiquetage initial.

Evolution de l'étiquetage : L'étiquetage initial est ensuite modifié itérativement en tenant compte de l'étiquetage des voisins de i . Pour cela, on utilise une notion de compatibilité $R_{ij}(z, z')$ mesurant la compatibilité pour que le point j soit dans $C_{z'}$ sachant que i est associé à la classe C_z . Pour chaque paire de classes z et z' , la matrice R de compatibilité est définie comme suit :

$$\begin{cases} R(z, z) = 1 \\ R(z, z') = 0 \text{ si } z \neq z' \end{cases} \tag{47}$$

On obtient le nouvel étiquetage par :

$$P_{i,n+1}(z) = \frac{P_{i,n}(z) [1 + Q_{i,n}(z)]}{\sum_{l=1}^c P_{i,n}(l) [1 + Q_{i,n}(l)]} \tag{48}$$

où

$$Q_{i,n}(z) = \frac{1}{\text{Card}(V_i)} \sum_{i' \in V_i} \sum_{z'} R(z, z') P_{i,n}(z') \quad (49)$$

$\text{Card}(V_i)$: représente le nombre de pixels du voisinage.

Grâce à ce processus, l'étiquetage du pixel i évolue en fonction de celui de ses voisins. Cette double contribution de i et son voisinage dans le nouvel étiquetage est traduite par les vecteurs $P_{i,n}$ et $Q_{i,n}$. En ce sens, $P_{i,n}$ peut être interprété comme l'avis du pixel sur son propre étiquetage et $Q_{i,n}$ comme l'avis de ses voisins.

Condition d'arrêt : Le processus est itéré jusqu'à ce qu'un nombre significatif de points soient bien affectés (exemple : leur vecteur de probabilité a une composante supérieure à 0.9).

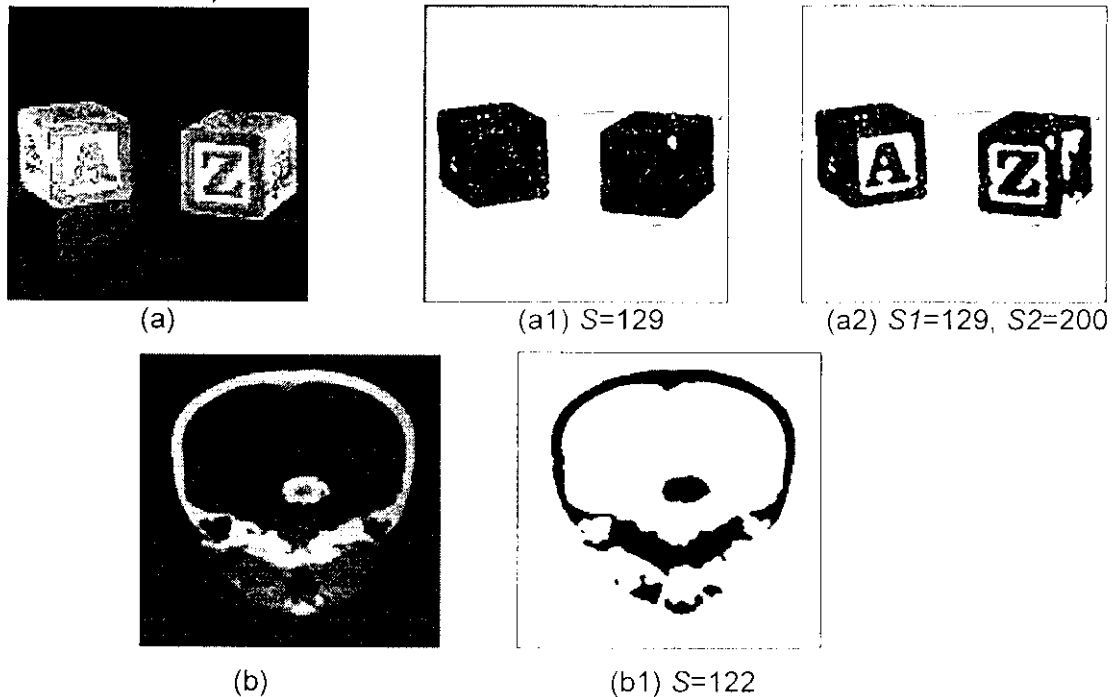


Figure 13 : Segmentation par la relaxation probabiliste.

L'image (a1) comporte les dés de l'image, séparation des objets de la scène, (a2) les dés avec leurs détails de forme, apparition des caractères A et Z, (b1) détection de la lésion.

Nous obtenons une bonne segmentation avec l'algorithme de relaxation, les objets obtenus sont lissés grâce au filtrage apporté par l'algorithme de relaxation. Ce qui permet d'avoir les contours des objets continus (non interrompus ou fragmentés) et la granulosité⁶ qui apparaît avec les techniques de seuillage a largement diminué. Mais notons que le temps d'exécution est plus important.

Notons que la qualité des résultats dépend énormément de l'étiquetage initial des points, donc le choix des bons seuils qui délimitent les classes est très important, pour cela, nous pouvons exploiter les algorithmes de Fisher et de seuillage automatique [87].

⁶ Points aléatoires qui apparaissent dans les objets

Dans la section suivante, nous proposons une technique de segmentation qui explore les propriétés statistiques de l'image, et produit l'image segmentée en se basant sur un principe de dichotomie hiérarchique de l'histogramme.

2.3.4 Algorithme de dichotomie hiérarchique d'histogramme

La segmentation d'image est le processus par lequel une image est divisée en un groupe de régions homogènes. Dans la technique de la dichotomie hiérarchique de l'histogramme, nous considérons l'image originale I de niveaux de gris comme étant l'union de régions homogènes non chevauchantes. La hiérarchie des régions de l'image est produite par la dichotomie d'histogramme. En effet, l'histogramme est à plusieurs reprises divisé en intervalles hiérarchiques continus tel que chacun ait une erreur carrée moyenne pixel par pixel (MSE) inférieure à un seuil T_σ .

L'histogramme MSE de l'intervalle de niveaux de gris $[d, u]$ est défini comme suit :

$$\sigma_{[d,u]}^2 = \frac{\sum_{k=d}^u P(k)(k-r_{[d,u]})^2}{\sum_{k=d}^u P(k)} \quad (50)$$

Où : d, u : sont les limites de l'intervalle.

La fonction $P(k)$ est l'histogramme normalisé en niveaux de gris $\left(\sum_k P(k)=1\right)$.

r est le niveau de gris quantifié de l'intervalle d'histogramme et est défini comme suit :

$$r_{[d,u]} = \frac{\sum_{k=d}^u P(k)k}{\sum_{k=d}^u P(k)} \quad (51)$$

Lorsque le MSE de l'intervalle d'histogramme est plus grand que T_σ , il sera divisé en deux sous-intervalles, tel que la somme des MSE soit réduite au minimum. Avec un point de division de l'intervalle $c_{[d,u]}$ choisi comme :

$$c_{[d,u]} = \arg \min \left\{ \sigma_{[d,c]}^2 + \sigma_{[c+1,u]}^2 \right\} \quad (52)$$

Avec cette méthode, un arbre hiérarchique des intervalles d'histogramme peut être obtenu, où chacun correspond à une ou plusieurs régions de l'image, qui ont une homogénéité interne approximative au sens du critère MSE minimal. Cette méthode est conforme à la quantification optimale d'image [88, 89] ; mais la hiérarchie de l'histogramme offre la flexibilité de commander la qualité de quantification, sans connaissance a priori du nombre de régions de l'image.

Par la figure suivante, nous présentons la segmentation de l'image des dés par l'algorithme de dichotomie d'histogramme DH pour un seuil $T_\sigma=75$.

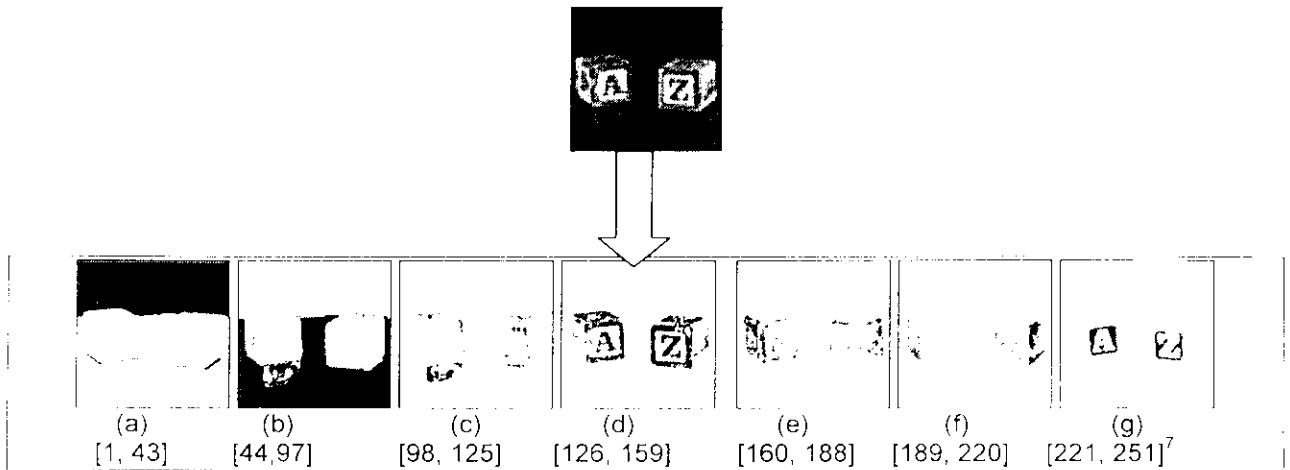


Figure 14 : Segmentation de l'image 2 dés par l'algorithme de DH pour $T_{\sigma}=75$.

Par les images segmentées obtenues, nous constatons que chacune correspond à une texture de l'image originale. En effet, l'algorithme a réussi à séparer le fond en : sa partie supérieure (Fig.14a), sa partie inférieure (Fig.14b) et l'ombre du dés de gauche (Fig.14c) malgré qu'ils ont trois textures difficilement distinguables. La fig.14g correspond au cadre des caractères A et Z des dés. Les images 14 (d, e, f) peuvent être rassemblées pour reconstituer les dés de l'image (Fig.15). Par les figures 16 à 18, nous présentons les principales classes lumineuses obtenues par l'algorithme DH appliqué sur différentes images.

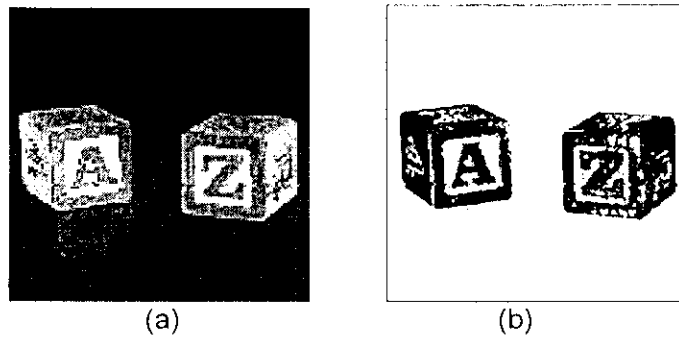
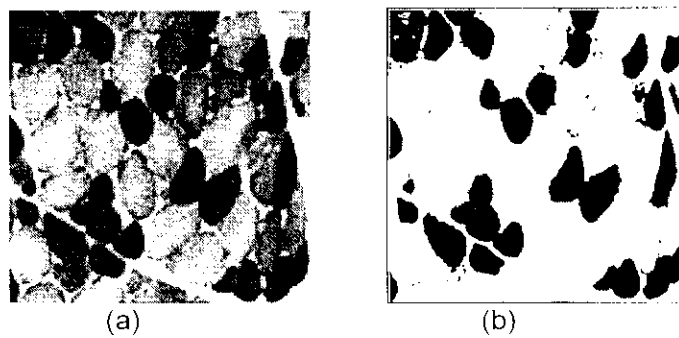


Figure 15 : Segmentation des dés par l'algorithme de DH.



⁷ La plage d'intensité de cette image est l'intervalle [0, 255].

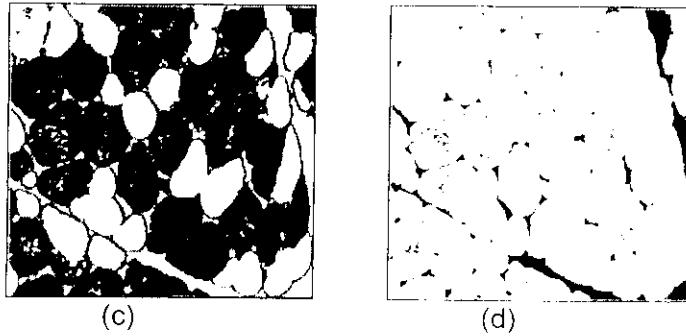


Figure 16: Segmentation de l'image muscle par l'algorithme de DH.

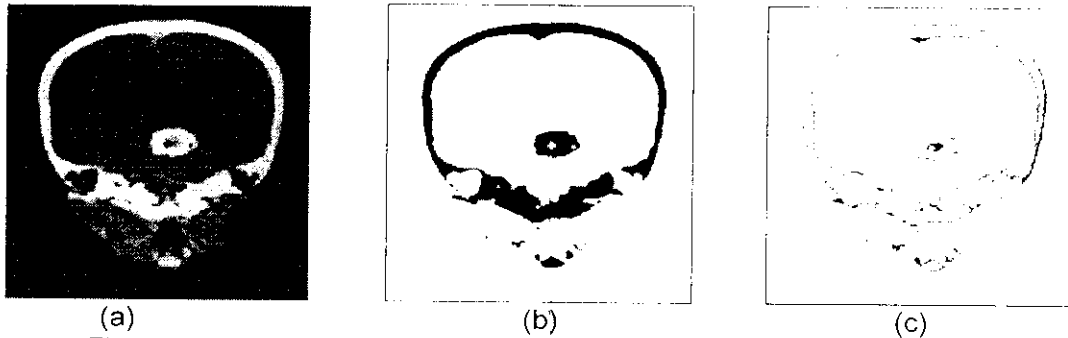


Figure 17 : Segmentation de l'image cerveau par l'algorithme de DH.

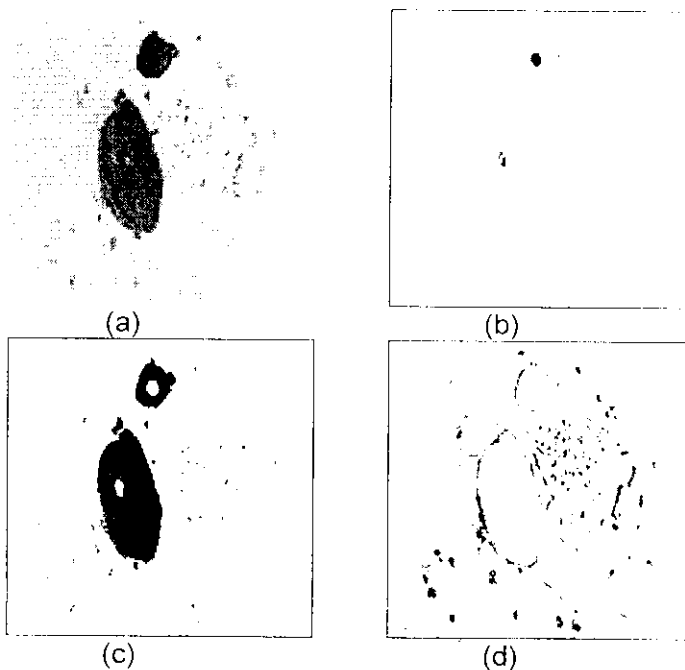


Figure 18 : Segmentation de l'image de cellules par l'algorithme de DH.

Ces images segmentées correspondent à des objets précis de la scène originale :

(15) Les dés ; (16b) la partie sombre du tissu musculaire, (16c) claire et (16 d) le fond de l'image ; (17b) séparation de la lésion du reste du cerveau et (17c) les contours de l'image cérébrale ; (18b) image des noyaux, (18c) image des cytoplasmes, (18d) comporte les zones non homogènes du fond.

Le tableau suivant présente les seuils T_i appliqués et les différents intervalles obtenus.

	T_0	$[d, u]$	Les images segmentées		
			(b)	(c)	(d)
Figure 15 (a)	75	[0, 43]			
		[44, 97]			
		[98, 125]			
		[126, 159]	*		
		[160, 188]	*		
		[189, 220]	*		
		[221, 251]			
Figure 16 (a)	75	[0, 82]	*		
		[83, 112]		*	
		[113, 142]		*	
		[143, 174]		*	
		[175, 208]		*	
		[209, 244]			*
Figure 17 (a)	75	[0, 68]			
		[69, 104]			
		[105, 135]	*		
		[136, 165]	*	*	
		[166, 196]	*		
		[197, 255]	*		
Figure 18 (a)	75	[1, 39]	*		
		[40, 72]		*	
		[73, 107]		*	
		[108, 142]		*	
		[143, 173]			*
		[174, 215]			
[216, 256]					

Tableau : Les intervalles obtenus par l'algorithme DH.

Le symbole "*" indique les classes lumineuses des images segmentées. Dans certaines images, pour détecter des objets bien précis, l'image segmentée doit correspondre à l'union de plusieurs intervalles d'intensité $\bigcup [d, u]$.

Notons que l'information spatiale de connectivité des pixels n'est pas considérée dans cet algorithme. Et pour améliorer son efficacité, les auteurs de [89, 90, 91, 92, 93] proposent d'appliquer une procédure d'optimisation avec les algorithmes génétiques après la segmentation.

2.3.5 Segmentation Multifractale

Le paramètre multifractal α est adapté à la segmentation en régions homogènes des images. Rappelons qu'un exposant de singularité mesure le degré de régularité et d'homogénéité d'une mesure, c'est cette caractéristique qui est exploitée pour la segmentation.

L'algorithme procède par la recherche et le groupement des points ayant des exposants de singularité proches, ainsi les différentes étapes de segmentation sont [6, 7] :

1. calculer les exposants de singularité en chaque point de l'image,
2. trouver la singularité minimale α_{min} et la singularité maximale α_{max} ,
3. diviser le spectre de singularité $[\alpha_{min}, \alpha_{max}]$ en N parties égales I_k ,
4. pour chaque sous intervalle I_k , trouver l'ensemble des points de l'image ayant l'exposant de singularité appartenant à l'intervalle I_k , ce qui nous permet d'avoir N sous images, chacune correspondant à un intervalle iso-alpha.

Notons que pour un intervalle $I_k = [\alpha_1, \alpha_2]$, les pixels retenus seront ceux dont le niveau de gris sera compris dans l'intervalle $\left[p\left(\frac{1}{L}\right)^{\alpha_1}, p\left(\frac{1}{L}\right)^{\alpha_2} \right]$:

Rappelons que le calcul de α à l'échelle du pixel s'effectue selon cette équation :

$$p(x,y) = p\left(\frac{1}{L}\right)^{\alpha(x,y)} \quad (53)$$

Où $p(x, y)$ est l'intensité du point (x, y) , p est la somme des niveaux de gris de toute l'image et L la taille de l'image.

La figure suivante présente les résultats de segmentation par l'algorithme multifractal. Les pixels noirs correspondent à ceux ayant des α dans le même intervalle I_k .

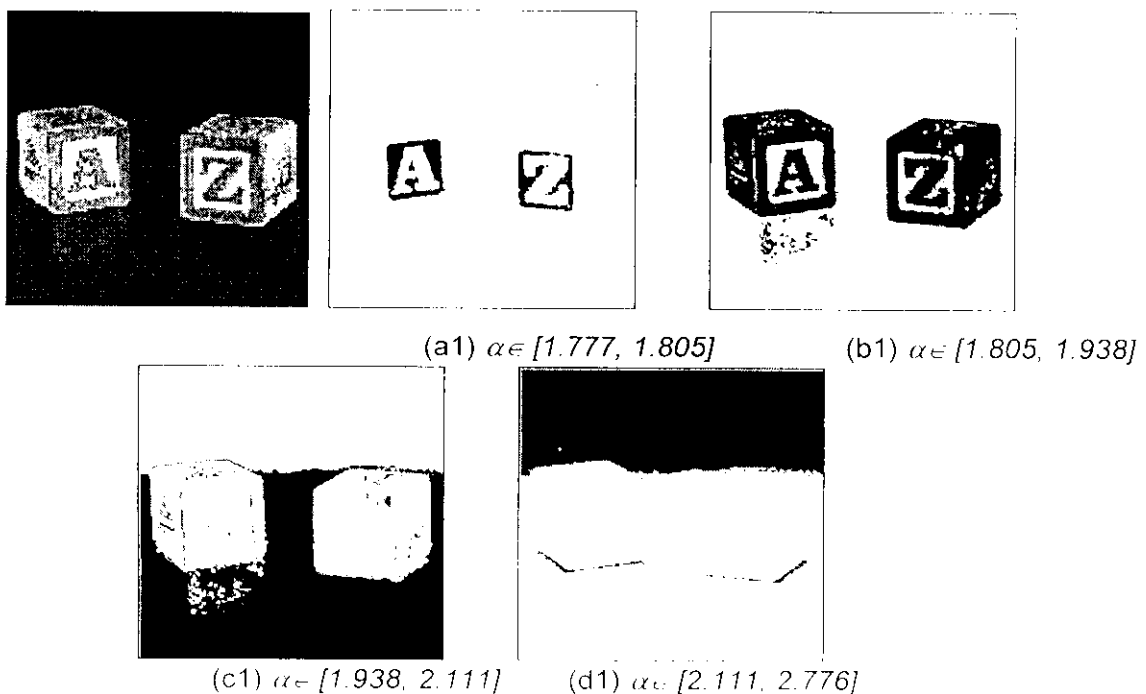


Figure 19 : Segmentation multifractale de l'image 2 dés. $\alpha_{Min}=1.777$, $\alpha_{Max}=2.776$

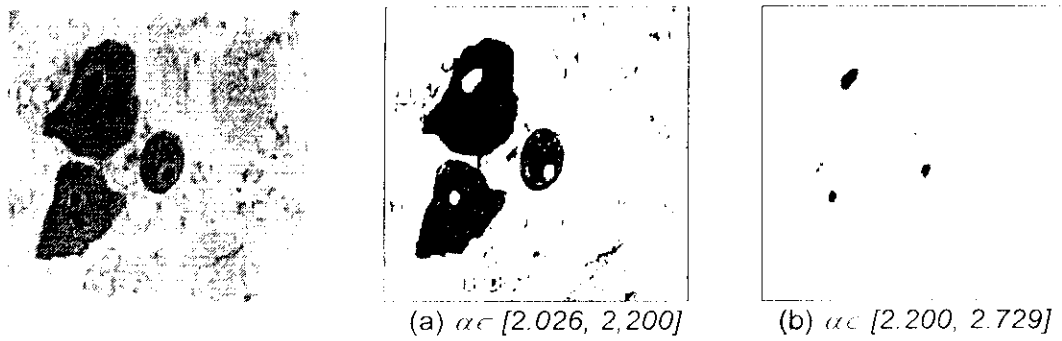


Figure 20 : Segmentation multifractale d'une image de cellules biologiques du col, $\alpha_{Min}=1.928$, $\alpha_{Max}=2.729$.

Par simple variation des intervalles de singularité, nous pouvons récupérer des objets bien précis de l'image originale. La segmentation effectuée est de texture et les textures claires correspondent aux basses valeurs de l'intervalle $[\alpha_{Min}, \alpha_{Max}]$, et vice versa.

Pour que la troisième étape de l'algorithme pour la recherche des intervalles I_k s'effectue d'une manière automatique, nous pouvons exploiter l'algorithme de seuillage automatique pour la recherche des intervalles iso-alpha [9]. Ceci est possible après échantillonnage de l'intervalle $[\alpha_{Min}, \alpha_{Max}]$ avec un pas fixe (par exemple de 0.01), et ensuite, par application de l'algorithme automatique du calcul des seuils sur l'intervalle $[\alpha_{Min}, \alpha_{Max}]$.

2.3.6 Segmentation par la morphologie mathématique

Nous allons présenter un algorithme de segmentation en régions homogènes basé sur un processus morphologique complexe de reconstruction géodésique.

La reconstruction géodésique utilise comme point de départ une image f appelée *image masque* et une image g appelée *image marqueur*. Dans le cas où $g \leq f$, la reconstruction consiste à appliquer jusqu'à stabilité et en chaque point p de l'image g , un ensemble de dilatations géodésiques élémentaires $\delta^j(f, g)(p)$ définies par [46, 64, 65] :

$$\delta^{(1)}(f, g)(p) = \text{Min}\{\delta_1(g)(p), f(p)\} \quad (54)$$

où $\delta_1(g)$ désigne la dilatation morphologique élémentaire utilisant un élément structurant de taille 1. Le processus de reconstruction est alors appelé *reconstruction par dilatation géodésique* et est défini par :

$$\rho(g \setminus f) = \delta^{(\infty)}(f, g) = \delta^{(1)}(\delta^{(1)}(\dots \delta^{(1)}(f, g) \dots)) \quad (55)$$

Dans le cas où $g \geq f$, une transformation duale de reconstruction par érosion géodésique est définie [46, 64, 65].

Un algorithme basé sur la reconstruction géodésique a été proposé [8, 94, 95, 96, 97] pour la segmentation, les différentes étapes sont :

- Ouverture de l'image originale par un élément structurant carré de taille presque égale à celle des objets à extraire.

- Reconstruction morphologique de l'image ouverte par rapport à l'image originale. Il s'agit d'appliquer des dilations géodésiques conditionnées à l'image originale, jusqu'à l'idempotence. Cette opération permet de récupérer la forme originale des objets de l'image ouverte, tout en gardant l'effet de filtrage morphologique de l'opération d'ouverture. Comme résultat, la forme des objets à extraire est intacte, alors que le reste de l'image est uniformément lumineux.
- Seuillage de l'image reconstruite. Notons que grâce au filtrage morphologique, le seuil est facile à choisir.

La segmentation de l'image des dés par l'algorithme de reconstruction géodésique morphologique donne les résultats suivants :

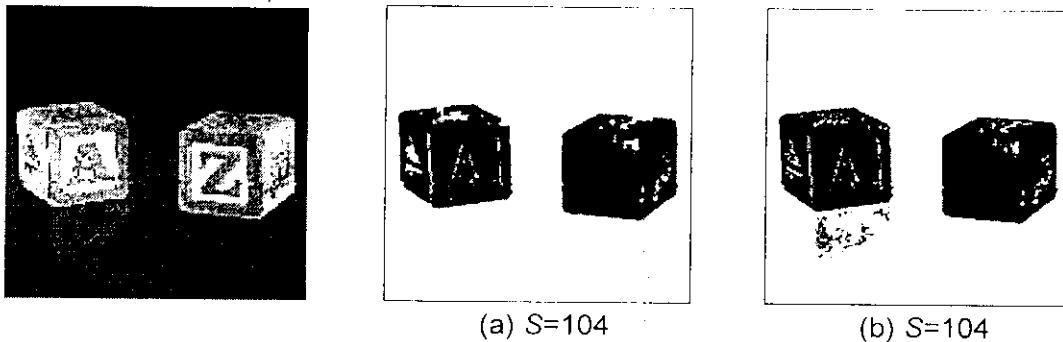


Figure 21 : Segmentation par l'opérateur morphologique géodésique.

En analysant la figure 21(a) qui présente la segmentation par reconstruction géodésique de l'image des dés, nous constatons la non apparition des points correspondant à l'ombre du dés de gauche, contrairement à l'image (21c) obtenue par un simple seuillage avec le même seuil.

Par la figure 22 nous présentons d'autres résultats de segmentation.

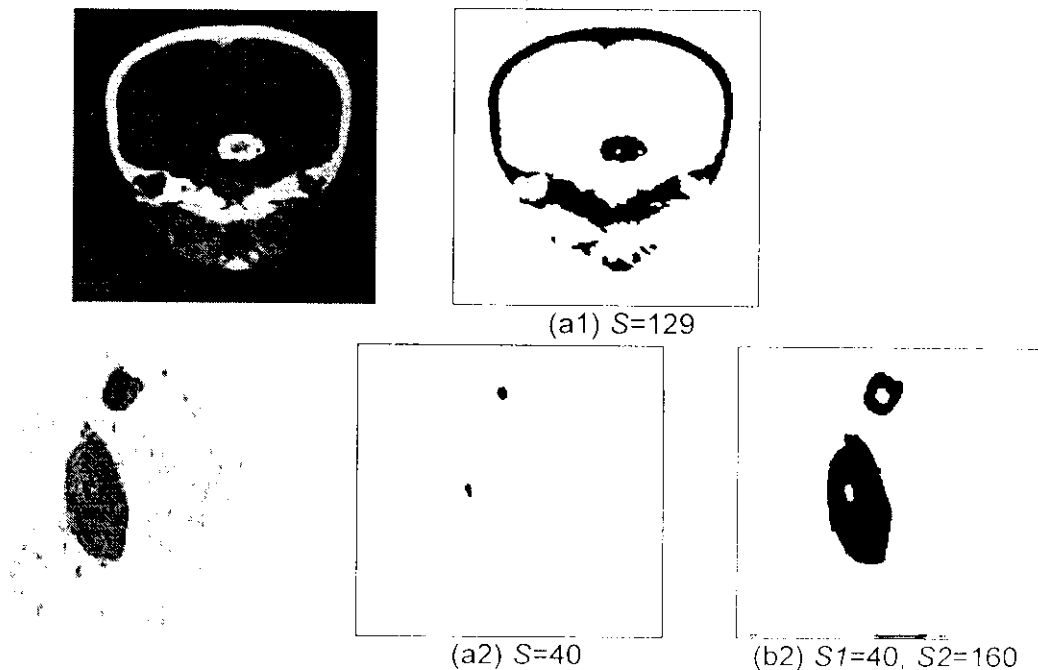


Figure 22: Segmentation par l'algorithme de reconstruction morphologique géodésique.

Sur l'ensemble des images, nous avons obtenu une bonne et nette segmentation des objets de la scène, puisque l'effet granuleux dû aux points aléatoires n'apparaît plus. Ceci est grâce au filtrage apporté par les opérateurs morphologiques qui permet d'uniformiser la texture des objets de l'image originale ce qui diminue l'apparition de ces points aléatoires dans les régions segmentées. Il permet aussi de rehausser une région lumineuse par rapport à sa voisine tout en respectant leurs limites, et c'est ce qui simplifie la recherche du seuil de binarisation.

Souvent après la binarisation, nous utilisons l'ouverture ou la fermeture à deux niveaux de gris avec un élément structurant carré de taille maximale 3x3, pour reboucher les trous (points blancs) ou éliminer les tâches noires autour des régions blanches.

Notons que l'élément structurant utilisé pour le calcul de la géodésie morphologique est de forme carrée et de taille choisie suivant l'objet à extraire de l'image. En effet, sa taille doit être proche de celle de l'objet à segmenter (la lésion, noyaux,...), à l'exception de la segmentation des cytoplasmes des images de cellules biologiques du col utérin.

En effet, les cytoplasmes et le fond possèdent des textures confondues, et ils se confondent d'avantage à cause du filtrage par les opérateurs morphologiques. D'où la segmentation des cytoplasmes sera une région qui comportent tous les cytoplasmes proches collés. Pour éviter ces erreurs, nous devons choisir un élément structurant de taille inférieure à la distance minimale entre deux cytoplasmes (généralement ne dépasse pas 5x5).

Cette méthode de segmentation est bien adaptée à la segmentation des lésions dans des images cérébrales [8, 97].

2.3.7 Conclusion

Nous avons étudié et appliqué ces techniques de segmentation en régions homogènes, les constatations suivantes ont été faites :

- ✓ Les techniques de seuillage s'avèrent non satisfaisantes pour une grande variété d'images à cause du manque d'utilisation de relations spatiales le long du processus de segmentation. Mais elles reviennent souvent, puisque dans la plupart des algorithmes (tel que la détection de contours), la binarisation est la dernière étape.
- ✓ La classification floue permet de moduler la décision finale d'étiquetage des points. Sachant que les points ayant un degré d'appartenance supérieur à un seuil sont qualifiés de sûrs, les autres sont classés ambigus. Un autre algorithme de segmentation récursive basé sur la LF 'Fuzzy C-Means' est étudié.
- ✓ Les méthodes de relaxation sont basées sur deux étapes : la classification initiale des pixels et le réajustement itératif des probabilités sachant que la probabilité de chaque pixel est ajustée suivant ses voisins. Cependant, la stabilité du processus est liée aux erreurs de calculs pouvant survenir du fait de la complexité des expressions de mises à jour. De plus, certains cas numériques particuliers (somme tendant vers zéro par exemple) ne sont pas à exclure. Il est nécessaire de définir une mesure explicite de la

consistance et de l'ambiguïté pour assurer une meilleure qualité de segmentation. Ces méthodes s'avèrent très coûteuses en temps de calcul et en place mémoire.

- ✓ La dichotomie d'histogramme est une technique de division qui procède par le groupement des points de l'image en utilisant un critère de mesure de texture : 'la variance', et ceci sans effectuer une partition préalable de l'image en sous-images, ce qui permet de respecter les limites des régions, ce qui n'est pas le cas des techniques classiques de division qui laissent des empreintes spécifiques dans la carte des régions avec des frontières marquées par de nombreux traits verticaux et horizontaux, ce qui occasionne parfois des régions particulièrement découpées.
- ✓ La segmentation par les exposants de singularité α permet la détection des différentes textures de l'image, puisque α est un paramètre de mesure de la texture. Cet algorithme a présenté de bons résultats, nous avons aussi proposé une amélioration de cet algorithme en exploitant l'algorithme de seuillage automatique pour la recherche des différents intervalles iso-alpha [9].
- ✓ L'algorithme de segmentation par la géodésie morphologique a donné de bons résultats. En effet, nous obtenons des régions connexes sans effet de granulosité grâce au filtrage morphologique qui simplifie aussi la recherche du seuil de binarisation.

Jusqu'à maintenant, nous avons présenté la segmentation qui permet un partitionnement de l'image et aussi de passer d'une description microscopique constituée de la liste des valeurs des niveaux de gris en chaque point, à une description macroscopique où chaque région mise en évidence peut être interprétée comme une composante de la scène analysée. En effet, après la répartition des niveaux de gris en classes selon la technique de segmentation choisie, les régions sont définies par les ensembles maximaux de pixels connexes appartenant à la même classe. Une autre forme de segmentation d'images en régions peut être appliquée, elle trouve son intérêt non pas dans l'extraction des classes, mais dans l'étiquetage optimal des points de l'image. La segmentation utilisée est celle dite par classification.

2.4 Segmentation par étiquetage en composantes connexes

L'étiquetage en composantes connexes d'une image attribue à tous les pixels d'une composante connexe une même étiquette (valeur entière) qui permet son identification.

Ces méthodes de classification se servent des relations de connexité pour déterminer les différentes régions, appliquées après la détermination de la partition de l'espace des luminances en utilisant les techniques de segmentation (contours ou régions).

Il existe plusieurs algorithmes d'étiquetage des régions connexes, nous présentons la méthode la plus classique qui repose sur un balayage séquentiel de l'image et la méthode de masquage récursif.

2.4.1 Algorithme d'étiquetage par balayage séquentiel de l'image

Une segmentation est d'abord effectuée pour identifier sur une image en plusieurs niveaux de gris, l'ensemble R de points-image connexes constituant une région. L'étiquetage peut être effectué sur une image binaire de régions homogènes ou de contours.

Si nous exploitons des images obtenues par détection de contours, pour lesquelles la région est identifiée par reconstitution de son contour fermé, nous pouvons obtenir la région par une procédure de remplissage [98]. Inversement, si nous exploitons des images obtenues par segmentation en régions homogènes, où la région est définie par sa fonction caractéristique, la frontière de cette région peut être obtenue par une procédure de *suivi de contour* [22].

Pour étiqueter les régions, A. Rosenfeld a proposé un algorithme qui a été adapté par J. F. Quesne et J. P. Cocquerez [22] afin de fonctionner en 4-connexité et d'étiqueter les régions dans leur ordre d'apparition conformément au balayage vidéo (de haut en bas et de gauche à droite) et l'ont proposé pour une architecture en temps réel pour l'étiquetage des régions.

Cette méthode détecte les adjacences entre pixels et définit l'étiquette du point courant en fonction de celles des points voisins. Elle nécessite deux phases : la première est un balayage pendant lequel on affecte une étiquette à chaque pixel, la seconde est la résolution des équivalences. En effet, au cours du balayage, une région peut avoir plusieurs étiquettes, par exemple, dans la figure 23, l'étiquette 3 doit être remplacée par 1. Or, on ne peut pas modifier toutes les étiquettes des points déjà à 3. L'algorithme utilise donc une table dans laquelle sont sauvegardées les équivalences entre étiquettes. Lorsque toute l'image a été traitée, il faut résoudre ces équivalences. L'image étiquetée est stockée, l'étiquette définitive étant obtenue par passage dans une table de look-up (LUT).

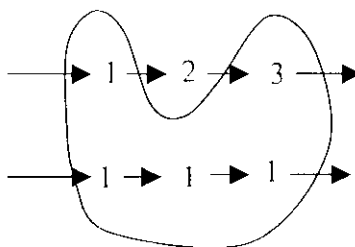


Figure 23 : Equivalence entre régions.

Nous présentons en détail cet algorithme, et avant d'entamer la démarche concernant le processus d'étiquetage, il est utile de définir les termes suivants :

Equivalence : Une étiquette 'a' est dite équivalente à l'étiquette 'b' s'il existe un chemin reliant le point correspondant à 'a' au point correspondant à 'b' dans l'image, tel que ce chemin ne passe pas par '0'.

Table d'équivalence : La réduction du nombre de balayage est mise en évidence grâce à cette table, du moment qu'elle permet de gérer les équivalences. La structure de son

implémentation est un tableau dynamique, sa taille augmente pendant le parcours séquentiel de l'image selon le besoin, c.-à-d., lors de l'apparition de nouvelles étiquettes et lors de l'insertion des étiquettes équivalentes à d'autres étiquettes.

Dans ce qui suit, nous présentons les deux phases du processus d'étiquetage.

2.4.1.1 La première phase : qui consiste en un balayage de l'image ligne par ligne, pendant lequel on affecte une étiquette temporaire à chaque pixel. A la fin de cette phase, une région peut avoir plusieurs étiquettes, ainsi l'algorithme doit utiliser la table d'équivalence dans laquelle sont sauvegardées les équivalences entre étiquettes. Nous procédons comme suit :

1. Initialisation : Soit *Ima* l'image binaire comportant les composantes connexes à étiqueter. Où les ensembles de pixels dont il faut extraire les composantes connexes sont à 1 et tous les autres pixels de *Ima* à 0⁸.

2. Premier balayage : Cette étape de l'algorithme procède comme suit :

- ✓ **Génération de la table d'équivalence** : Au fur et à mesure qu'on progresse dans le balayage de l'image⁹, de nouvelles étiquettes apparaissent, par conséquent on alloue une nouvelle colonne pour chacune d'elles. Nous considérons pour chaque point *s* ses voisins déjà traités. Dans notre cas, nous travaillons avec la 4-connexité. Donc, pour chaque point, le nombre des prédécesseurs avec la 4-connexité est deux.
- ✓ **Mise à jour de la table d'équivalence** : Elle se fait de la manière suivante :

Lors de ce premier balayage de l'image et en arrivant à un pixel à étiqueter dont les prédécesseurs sont différents, on prend la dernière étiquette de la colonne (et cela au niveau de la table d'équivalence) correspondant à l'étiquette de l'un des deux prédécesseurs, soit 'a' cette étiquette. Ensuite, on récupère la dernière étiquette de la colonne correspondant à l'étiquette de l'autre prédécesseur, soit 'b' cette étiquette. Après une comparaison entre 'a' et 'b', l'étiquette la plus basse est attribuée au pixel courant (C'est l'étiquette *min1*, voir le pseudo code), elle est aussi insérée dans la colonne de l'étiquette de l'autre prédécesseur.

Une chose importante est à signaler concernant l'insertion : cette dernière se fait selon l'ordre décroissant, c.-à-d., que les étiquettes d'une même colonne (exception faite pour la première étiquette) sont classées de la plus grande à la plus petite, ce qui nous conduit à dire que la dernière étiquette d'une colonne donnée n'est rien d'autre que la plus petite étiquette équivalente à la première étiquette se trouvant dans cette même colonne.

L'algorithme du premier balayage est donné par ce pseudo-code :

⁸ Si la configuration initiale de l'image binaire n'est pas ainsi, procéder par inversion de *Ima*.

⁹ Comme solution au problème de l'effet de bord, les éléments de la première ligne et de la première colonne sont traités isolément, sauf le pixel de coordonnées (1,1) qui reste tel qu'il était dans l'image binaire.


```

Début
etiq := 1;
Pour toutes les lignes de lma faire
    Tant que non fin de la ligne courante faire
        si (lma[i,j] < > 0) alors
            si (tous les prédécesseurs de lma[i,j] sont à zéro) alors
                lma[i,j] := etiq ;
                etiq := etiq ++
            sinon
                si (tous les prédécesseurs de lma[i, j] ont la même étiquette 'e')
                    alors lma[i,j] := e
                    sinon
                        e1 = min1;
                        lma[i,j] := e1 ;
                        mise à jour de la table d'équivalence.
                    Fsi
                Fsi,
            FTant,
        Fpour,
    Fin

```

2.4.1.2 La deuxième phase : Elle comporte la résolution des équivalences et un deuxième balayage de l'image pour l'attribution aux pixels des étiquettes définitives.

1. Résolution des équivalences : Elle consiste à assigner à chaque étiquette, l'étiquette la plus petite qui lui est équivalente, et cela va se faire au niveau de la table d'équivalence.

L'étape de la résolution des équivalences est effectuée selon l'algorithme suivant :

```

Début
    Pour i := 1 à max_ligne faire
        Pour j := 1 à max_etiq faire
            Début
                si T[i+1,j] < > 0 alors
                    Pour k := 1 à max_etiq faire
                        si ( T[1,k] = T[1,j] ) alors T[1,k] := T[i+1,j] ;
                    Fpour,
                Fpour,
            Fin.

```

avec : *max_ligne* : est l'indice de la dernière ligne de la table d'équivalence.

max_etiq : la plus grande étiquette.

Ces deux nombres ne sont connus qu'après la fin du premier balayage.

Notons qu'à la fin de cette étape, les différentes étiquettes à affecter aux pixels de l'image lma se trouvent sur la première ligne de la table d'équivalence.

2. Deuxième balayage : C'est la dernière phase du processus d'étiquetage. Ce second balayage de l'image exploite la table d'équivalence *T* pour affecter à chaque pixel l'étiquette finale qui lui est équivalente. Cette phase s'effectue selon cet algorithme :

Début*Pour toutes les lignes de lma faire**Tant que non fin de la ligne courante faire**Début**si ($lma[i,j] < > 0$) alors $lma[i,j] := T[1,lma[i,j]]$;**Fin**Fin*

où $T[i,j]$ est l'étiquette de la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne dans la table d'équivalence.

Notons que cet algorithme repose essentiellement sur la table d'équivalence, et vu l'intérêt majeur qu'elle présente, nous devons la gérer soigneusement car sa mauvaise gestion conduit inévitablement à un mauvais étiquetage. Dans ce qui suit, nous présentons un autre algorithme d'étiquetage qui ne repose pas sur la gestion d'une telle table d'équivalence.

2.4.2 Algorithme de masquage récursif [99]

Cet algorithme est d'autant plus intéressant qu'il donne une information supplémentaire concernant les frontières des régions. En effet, il permet d'obtenir simultanément en un seul passage sur l'image l'étiquetage des régions et l'extraction des frontières des régions. Il constitue en fait, une méthode coopérative entre suivi de contour et étiquetage.

On appelle fonction caractéristique de cette région R , une fonction égale à l'unité pour tous les points de R et nulle en dehors. Elle représente le masque obtenu pour une classe donnée.

Disposant d'une image binaire qui regroupe une ou plusieurs régions connexes, l'algorithme de suivi de contour est utilisé pour résoudre le problème de l'étiquetage complet des régions en effectuant un seul passage sur l'image.

Nous définissons un point d'intérêt comme étant un point dont la fonction caractéristique est à un. Nous scrutons l'image ligne par ligne jusqu'à rencontrer un point d'intérêt. Nous affectons à ce point l'étiquette 1 et le classons comme étant un point contour. A partir de ce point initial (x_0, y_0) , nous effectuons un suivi de contour jusqu'à atteindre le point de départ. Chaque point rencontré pendant le suivi recevra l'étiquette du point (x_0, y_0) et sera classé comme point contour. On passe alors au point suivant à partir du point initial.

A un moment donné du parcours de l'image, pour incrémenter le compteur d'étiquettes, nous devons obligatoirement rencontrer un point qui vérifie les conditions suivantes :

- Point d'intérêt,
- Point non-contour,
- Point non-étiqueté,
- Point non connexe à un contour,
- Point non connexe à un point étiqueté.

Si un tel point est rencontré, on lui affecte une nouvelle étiquette et on effectue un suivi de contour à partir de ce point qui devient le point initial (x_0, y_0) . Les points rencontrés pendant le suivi de contour auront la même étiquette que le point initial et seront classés comme points contour

Si pendant le parcours du suivi un autre point d'intérêt étiqueté différemment est rencontré, on arrête le suivi et tous les points étiquetés pendant le suivi seront étiquetés avec la même étiquette du point rencontré et ne seront pas classés comme points contour. On passe alors au point suivant à partir du point initial (x_0, y_0) après avoir décrémenté le compteur d'étiquettes d'une unité.

Si un point d'intérêt rencontré est connexe à un point déjà étiqueté, il reçoit l'étiquette de ce point et sera classé non-contour.

Si le point rencontré est connexe à un point contour, il reçoit l'étiquette du contour et sera classé non-contour. Un deuxième test est alors effectué pour voir si un autre point non-contour lui est connexe. Si c'est le cas, ce point aura la même étiquette que le contour et on refait à partir de ce point la même opération jusqu'à ce que toute la chaîne des points connexes ait la même étiquette que le point contour rencontré.

Par la figure 24, nous présentons deux exemples d'étiquetage, et par la figure 25, nous donnons l'organigramme de l'étiquetage par masquage récursif.

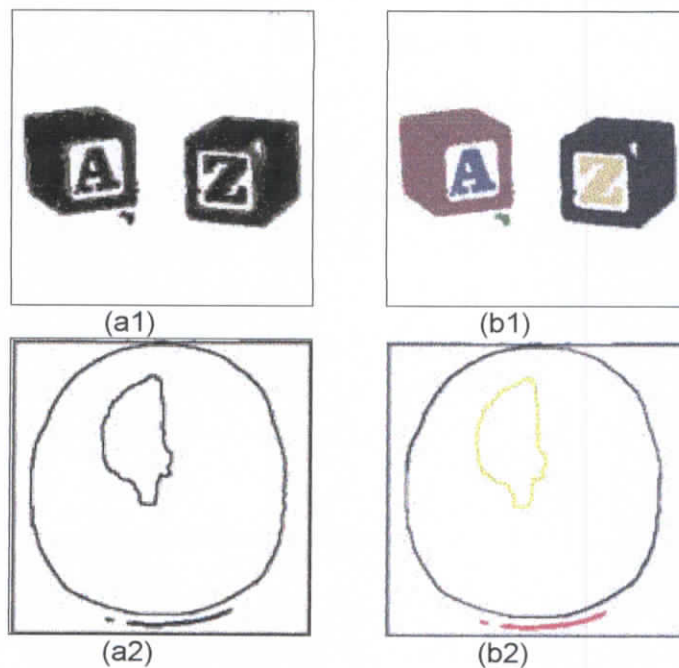


Figure 24 : Etiquetage des objets des images.

Par la figure (24.b1), nous avons un bon étiquetage des différents objets de la scène, même pour ceux qui ont à l'intérieur des zones vides ou d'autres objets (par exemple les alphabets dans les dés). Ce qui est très important, car en effet, certains algorithmes [57] peuvent découper le même objet en plusieurs régions à cause des vides qu'il comporte à l'intérieur. L'étiquetage des contours est donné par la figure 24.b2.

L'algorithme de masquage récursif est plus intéressant que l'algorithme de Rosenfeld puisqu'il effectue l'étiquetage en une seule passe de l'image sans gestion de table d'équivalence.

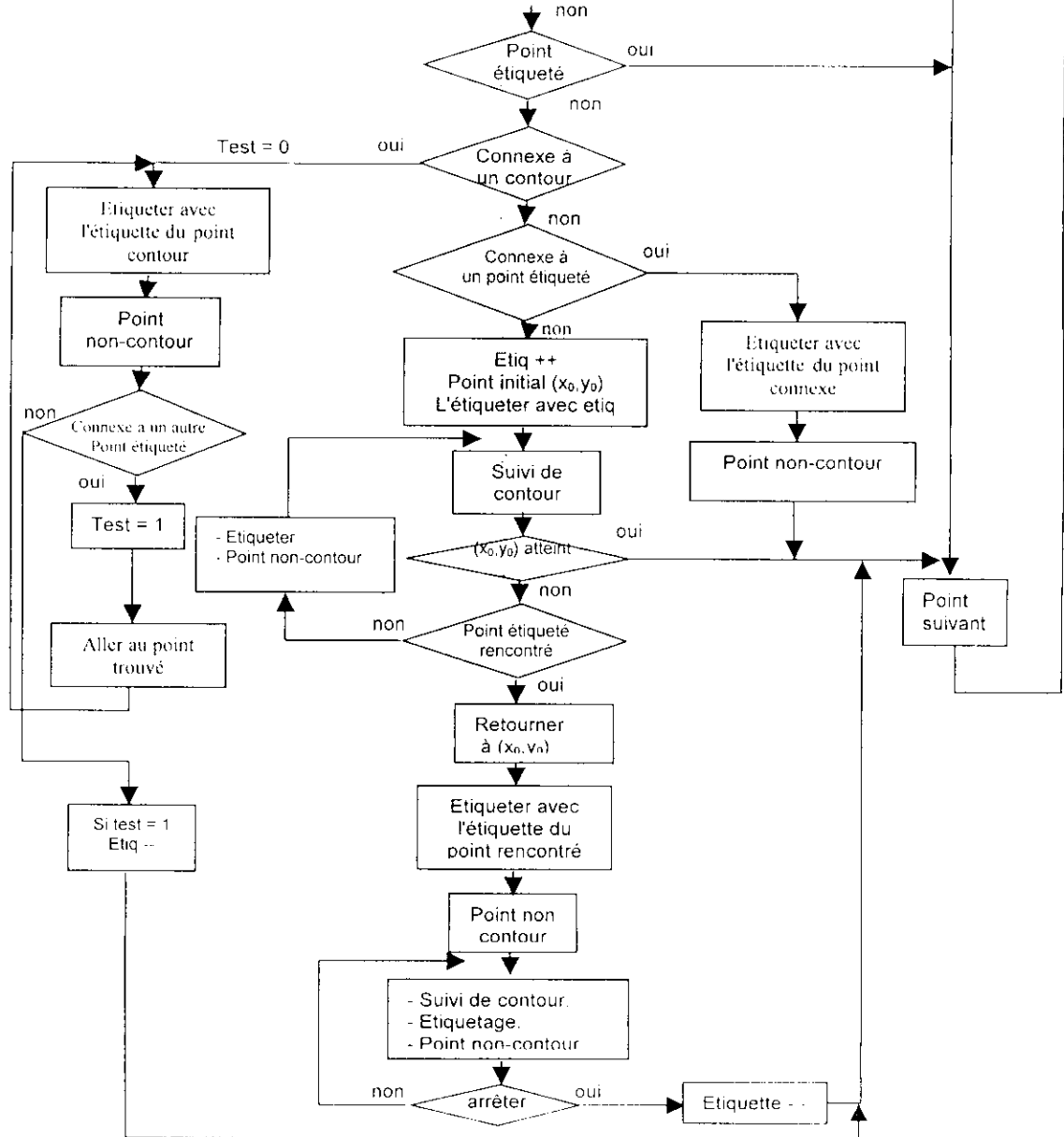


Figure 25 : Organigramme de la méthode d'étiquetage par masquage récursif.

segmentation en régions homogènes plutôt que pour la détection de contours. En effet, avec ces derniers, il est souvent difficile d'adapter la segmentation obtenue avec le besoin de l'application (reconnaissance), puisque souvent les contours obtenus ne correspondent pas aux limites des objets de la scène et sont souvent non fermés, discontinus. Et pour des images très complexes, telles que les images de cellules biologiques où les limites des corps cellulaires sont floues et mal définies, la détection est encore plus difficile.

Avec la segmentation multifractale en régions homogènes, nous obtenons de bons résultats qui sont valables et exploitables pour la reconnaissance afin de dépister les anomalies [6, 7]. En effet, l'algorithme multifractal permet la détection des points aléatoires qui définissent la granulosité nucléaire et cytoplasmique, qui est un critère de présence de la maladie. Le calcul du nombre de ces points dans le noyau (ou cytoplasme) par rapport à la surface du noyau (ou cytoplasme) permet d'estimer cette granulosité au niveau de chaque corps cellulaire. Cependant, les techniques qui introduisent un filtrage¹⁰ atténuent ces points essentiels à l'estimation de la granulosité, et les paramètres de texture (moyenne, variance, ...) les détectent difficilement, puisqu'ils opèrent sur un voisinage, mais non au niveau du pixel.

Malgré la sensibilité des exposants de singularité aux textures des noyaux et des cytoplasmes et à la détection de la granulosité, la séparation du cytoplasme du fond n'est pas toujours évidente, surtout s'il apparaît des régions de texture très proche de celle du cytoplasme et qui se collent ou se touchent aux cellules. Par la segmentation, nous obtenons une difformité des limites des cytoplasmes, et c'est une source d'erreur pour l'étape de reconnaissance où les formes des corps cellulaires sont analysées pour détecter la maladie.

Pour remédier à cet inconvénient, dans la suite de notre travail, nous explorons une autre théorie qui a trouvé des applications dans divers domaines : Les algorithmes génétiques.

¹⁰ Souvent, il s'agit des méthodes qui ne travaillent pas à l'échelle du pixel mais traite chaque point avec son voisinage

Chapitre 3

Les Algorithmes Génétiques : Théorie et Applications en Vision

- 3.1 Introduction
- 3.2 Fondements des algorithmes génétiques
 - 3.3 Les paramètres d'un AG
 - 3.4 Les étapes d'un AG
 - 3.5 Les variantes du modèle génétique
- 3.6 Association des AGs avec des méthodes locales
 - 3.7 Le parallélisme
- 3.8 Applications des Algorithmes Génétiques en Vision
 - 3.9 Conclusion

Les Algorithmes Génétiques :

Théorie et Applications en Vision

3.1 Introduction [11, 12, 13, 14, 100, 101]

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines des caractéristiques de l'exploration humaine. A chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente. Bien qu'utilisant le hasard, les AGs ne sont pas purement aléatoires. Ils exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance. Ils ont été développés vers les années 60 par John HOLLAND, ses collègues et étudiants à l'université du Michigan, dans le but de concevoir des systèmes artificiels qui possèdent des propriétés des systèmes naturels.

Ayant été reconnu comme une approche valide des problèmes nécessitant une exploration performante, et économique du point de vue calcul, les AGs sont maintenant appliqués plus largement aux domaines d'affaires, à la recherche scientifique,...

Ajouté à cela, les AGs risquent moins d'être piégés dans des minima locaux comme les algorithmes classiques (recuit simulé, recherche tabou,...) du fait qu'ils explorent, en parallèle, un ensemble de solutions possibles des problèmes traités.

Les AGs qui sont très utilisés pour résoudre les problèmes d'optimisation, diffèrent des autres méthodes selon quatre axes qui sont :

- Ils utilisent un codage des paramètres, et non les paramètres eux-mêmes.
- Ils travaillent sur une population¹ de points, au lieu d'un point unique.
- Ils n'utilisent que les valeurs de la fonction étudiée ; pas sa dérivée ou une connaissance auxiliaire.
- Ils utilisent des règles de transition probabilistes et non déterministes.

Il est donc clair qu'un AG manipule un ensemble d'individus (solutions) à la fois, où chaque individu est caractérisé par une empreinte génétique, c.-à-d., un ensemble de chromosomes. L'évolution des AGs se fait par deux mécanismes : la sélection naturelle et la reproduction ; la sélection naturelle fait que seuls les individus les mieux adaptés survivent et se reproduisent ;

¹ L'ensemble des individus traités simultanément par l'algorithme génétique constitue une *population*

quant à la reproduction sexuée, elle assure un brassage de l'ensemble des solutions. Ce phénomène (sélection/reproduction) réitéré plusieurs fois sur une population nombreuse produit après plusieurs générations des individus très bien adaptés au milieu dans lequel ils vivent.

Dans le cadre de ce chapitre, nous présentons les AGs : les notions de base, les champs d'applications et plus particulièrement les travaux effectués en vision.

3.2 Fondements des algorithmes génétiques

Les AGs, qui sont une transcription directe du processus naturel d'évolution au monde de l'informatique, sont fondés sur :

- Une représentation chromosomique des solutions du problème.
- Une méthode pour générer une population initiale de solutions.
- Une méthode d'évaluation qui classe les solutions selon leurs aptitudes.
- Des opérateurs génétiques, qui définissent la manière dont les caractéristiques des parents sont transmises aux descendants.

3.2.1 La représentation chromosomique

C'est un moyen de coder les solutions d'un problème ou encore les individus. Chaque solution est constituée d'une chaîne de symboles appartenant à un alphabet à priori de petite taille. Le symbole a un rôle dans les AGs similaire au rôle du gène dans la génétique naturelle. En général, chaque individu (solution) est représenté par une chaîne binaire de 0 et 1. Le choix d'un code doit tenir compte de la complexité du processus de codage/ décodage qui peut ralentir les calculs et influencer considérablement la convergence de l'AG.

Historiquement, le codage utilisé par les AGs était présenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état [11, 12, 13, 14]. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation simples. C'est également en utilisant ce type de codage que les premiers résultats de convergence théorique ont été obtenus.

Cependant, ce type de codage n'est pas toujours bon comme le montrent les deux exemples suivants :

- Deux éléments voisins en terme de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray.
- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du chromosome n'a pas alors forcément la structure du problème.

Les AGs utilisant des vecteurs réels évitent ce problème en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire [11, 12, 13, 14]. La structure du problème est conservée dans le codage.

Il existe deux types de difficultés pour le choix du codage. D'une part celui-ci doit pouvoir être adapté au problème de façon à limiter au mieux la taille de l'espace de recherche (espace de solutions), et aussi de façon que les nouveaux individus engendrés par les opérateurs génétiques (croisement, mutation, ...) soient souvent les plus significatifs ; c.-à-d., qu'ils puissent coder des solutions valides en respectant les contraintes du problème à résoudre.

Le choix d'un codage approprié est un élément critique dont dépend grandement l'efficacité d'un AG [11, 100, 101].

Un individu est défini par son **génotype**, c.-à-d., une structure de données. Ce génotype est exprimé par la fonction d'évaluation pour obtenir son **phénotype**. Celui-ci indique l'aptitude de l'individu à substituer dans son environnement (à se reproduire).

Le génotype est constitué d'un ensemble de gènes. Chaque gène peut prendre un certain nombre de valeurs dénommées **allèles**. Pour une représentation binaire par exemple : les gènes sont les bits et les allèles auront pour valeurs 1 et 0.

3.2.2 La population initiale

Les AGs manipulent un ensemble de solutions à la fois ; pour cela, on doit avoir au départ une population initiale de solutions, celle-ci est générée soit d'une façon aléatoire, soit à l'aide d'autres méthodes d'optimisation ou encore à base d'heuristique spécifique au problème.

Le choix de la population initiale d'individus conditionne fortement la rapidité de convergence de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état en veillant à ce que les individus produits respectent les contraintes. Si des informations a priori sur le problème sont disponibles, il paraît bien évidemment naturel de générer les individus dans un sous-domaine particulier afin d'accélérer la convergence.

3.2.3 La fonction d'évaluation

Elle prend en argument l'individu et lui associe un coût de performance. Cette fonction doit guider l'algorithme vers l'optimum en réalisant implicitement une pression de sélection dans cette direction. De plus, l'efficacité d'un AG s'appuie pour une grande part, sur la qualité de cette fonction. La mise au point d'une bonne fonction d'adaptation doit respecter plusieurs critères qui se rapportent à sa complexité et à la satisfaction des contraintes du problème. Lorsque la fonction d'adaptation apparaît excessivement complexe, consommant une importante puissance de calcul, la recherche d'une approximation plus simple est souhaitable.

3.2.4 Les opérateurs génétiques

Ces opérateurs sont la base des AGs, définissent la manière dont les individus se recombinent et s'agencent pendant la phase de reproduction. Les opérateurs qu'on retrouve le plus souvent sont : sélection, croisement et mutation.

3.2.4.1 Sélection

Cette opération est fondée sur le principe d'adaptation de chaque individu d'une population à son environnement, suivant la théorie de la sélection naturelle introduite par Charles Darwin. Ainsi, seuls les individus les plus aptes à répondre à certains critères seront sélectionnés à survivre et à se multiplier. Une fonction d'évaluation de l'adéquation des individus est associée au problème. Elle évalue chaque chromosome de la population, et seuls ceux dont la qualité sera suffisante seront conservés d'une génération à une autre. D'autres individus prendront la place des plus mauvais.

N' individus seulement sur N ($N > N'$) sont remplacés dans la population pour obtenir la nouvelle génération, N : étant la taille de la population. Le nombre de remplacements ayant été fixé arbitrairement par l'utilisateur. Les $(N - N')$ individus qui survivent sont parmi les meilleurs. Il s'agit d'une stratégie dite *élitiste*². Les N' autres individus de la génération suivante sont obtenus suite à une sélection en fonction de la valeur d'adaptation de chaque individu et application des opérateurs de reproduction.

En effet, lors de cette phase, une population temporaire contenant des copies conformes des parents (individus de la population courante) est formée, où chaque parent contribue par un nombre de copies en accord avec sa performance ; les individus les plus adaptés tendent à donner un plus grand nombre de copies que les moins adaptés ; ainsi la phase de sélection spécifie les individus de la population qui doivent survivre et auxquels seront appliqués les opérateurs de croisement et mutation.

Dans ce qui suit, nous présentons une procédure classique de sélection :

Soit une population de N individus, la sélection de chaque individu i dépend de sa probabilité P_{Si} , appelée probabilité de sélection. Dans le cas d'un problème de maximisation (recherche d'un maximum de la fonction d'adaptation), P_{Si} est donné par la formule :

$$P_{Si} = f_i / \left(\sum_{j=1}^N f_j \right) \quad (1)$$

où : f_i, f_j : sont respectivement les valeurs de la fonction d'évaluation pour les individus i et j .

Dans le cas de minimisation :

$$P_{Si} = 1 - \left(f_i / \left(\sum_{j=1}^N f_j \right) \right) \quad (2)$$

L'opérateur de sélection peut être mis en œuvre sous forme algorithmique de différentes façons. La plus simple est de créer une roue de loterie (Figure1) pour laquelle le $i^{\text{ème}}$ individu de la population courante occupe une section de la roue proportionnelle à son adaptation.

² Une stratégie *elitiste* consiste à conserver dans la population, d'une génération à l'autre, au moins l'individu ayant la meilleure adaptation

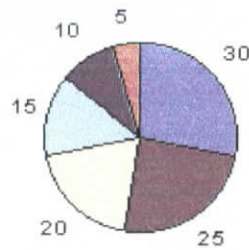


Figure 1 : Roue de Loterie

Les étapes de la sélection sont :

- Calcul de f_i pour chaque individu i de la population courante.
- Calcul de Ps_i de chaque individu i .
- Calcul de la probabilité cumulative $Ps_i = \sum_{j=1}^i Ps_j$ pour chaque individu i .
- Tourner la roue de loterie autant de fois qu'on a besoin de descendants (en général n fois, pour garder le même nombre d'individus dans la population).

Le principe est donné par l'algorithme suivant :

```

Pour  $i=1$  jusqu'à  $N$  faire
  Générer aléatoirement un nombre réel  $R_i$ , tel que  $R_i \in [0, 1]$ .
  Si  $R_i < Ps_1$  alors
    Sélectionner l'individu d'indice 1.
  Sinon
    Sélectionner l'individu d'indice  $k$ ,  $2 \leq k \leq N$ 
    Tel que :  $Ps_{k-1} < R_i \leq Ps_k$ 
  Fsi
Fpour
    
```

La méthode de la roue de loterie, telle qu'elle est représentée, peut avoir des problèmes dans le cas où un individu qui n'est pas forcément un optimum global a une section de surface importante sur la roue de loterie. Ce phénomène augmente la probabilité que le processus d'optimisation se bloque sur un optimum local et il ne permet pas d'exploiter d'autres régions de l'espace de recherche. En effet, la variance de cette méthode est élevée, par malchance, il est possible à la limite qu'un individu de mauvaise qualité soit sélectionné pour la reproduction autant de fois qu'il y a d'individus à remplacer. Il est aussi possible qu'un individu ayant une bonne valeur d'adaptation ne soit jamais sélectionné. Ce phénomène est responsable de la *dérive génétique* qui permet à certains individus de « survivre » au détriment d'individus meilleurs. Pour limiter ce risque, la taille de la population doit être suffisamment grande.

Une des méthodes proposée pour résoudre ce problème est appelée *méthode du rang*. Elle consiste à ordonner les individus en fonction de leurs performances (meilleure adaptation)

par ordre décroissant. Puis, on ordonne à chaque individu une probabilité de sélection relative à son rang au lieu de son évaluation. Ainsi, Ps_i de chaque individu i est :

$$Ps_i = \frac{N - rang_i + 1}{\sum_{j=1}^N rang_j} \quad (3)$$

Où : N est le nombre d'individus dans la population,

$rang_i, rang_j$: sont respectivement les rangs des individus d'indice i et j .

L'exemple suivant (Figure 2) montre comment les surfaces sont divisées dans le cas de la sélection basée sur la valeur d'adaptation et le cas de la sélection basée sur le rang.

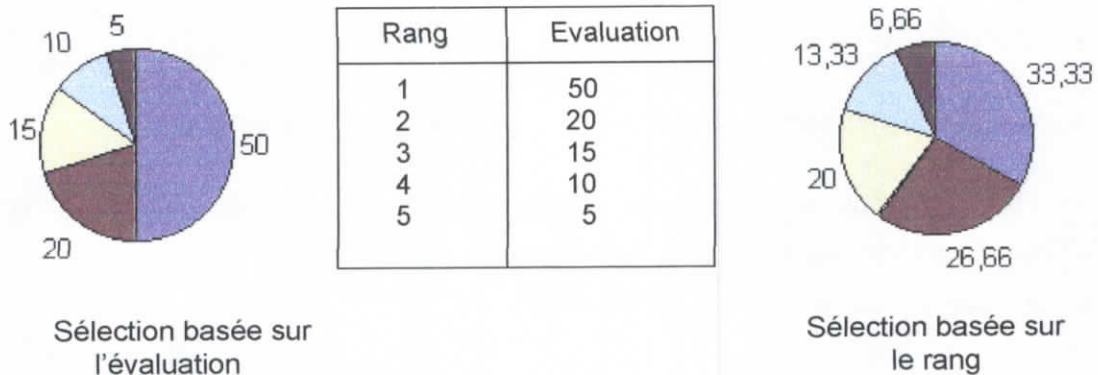


Figure 2 : Exemple de probabilités de sélection basée sur l'évaluation et le rang.

En fin de la sélection, on aura une nouvelle population appelée population intermédiaire (P_{int}) qui est constituée des individus ayant les bonnes valeurs d'adaptation qui sont sélectionnés plusieurs fois, et les mauvais vont disparaître de la population.

Pour créer les chromosomes de la nouvelle population, on fait appel aux mécanismes de croisement et de mutation.

3.2.4.2 Croisement [11, 12, 13, 14, 100, 101, 102]

Le croisement consiste à combiner deux individus quelconques (dits parents) pour en ressortir deux autres individus (dits enfants) pas forcément meilleurs que les parents ; il agit donc sur une paire d'individus pour donner une autre paire d'individus.

Il existe plusieurs variantes de cet opérateur, mais en général il consiste à couper en un ou plusieurs points deux individus (aux mêmes endroits dans les deux individus) et à échanger les parties situées entre ces points, ces derniers sont générés aléatoirement.

Le croisement sera appliqué à la population intermédiaire (P_{int}) résultant de la sélection. De ce fait, à chaque génération $N \times Pc$ individus subissent en moyenne un croisement, où N est la taille de la population et Pc le taux de croisement. Le processus se présente comme suit :

Pour chaque individu de la population
 Générer un réel $r \in [0, 1]$
 Si $r < Pc$ alors l'individu est marqué pour un croisement

On distingue plusieurs variantes de croisement :

Le croisement 1-point : Chacun des deux individus d'une paire est divisé en deux parties dont l'une des deux sera échangée avec la même partie de l'autre individu.

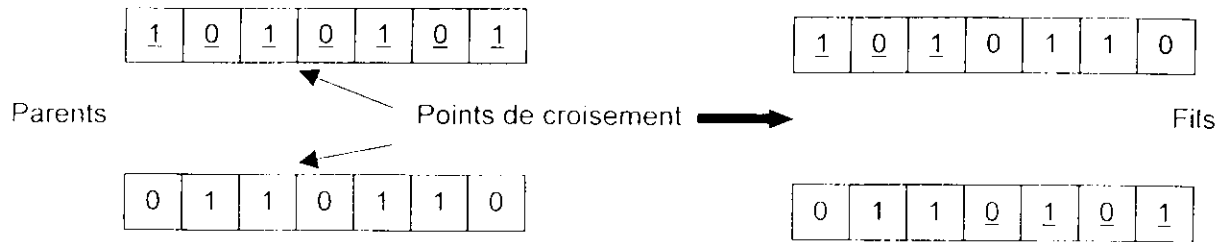


Figure 3 : Exemple du croisement à un point

Le croisement 2-points : Les individus sont coupés en deux points et les parties centrales des deux individus sont échangées.

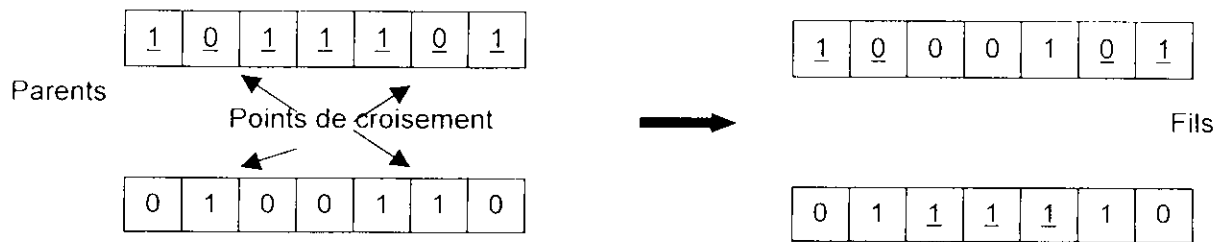


Figure 4 : Exemple du croisement à deux points

Le croisement N-points : Est une généralisation du croisement 2-points où les individus sont coupés en N-points où un tronçon sur deux est échangé.

Le croisement uniforme : Ce croisement utilise une chaîne de bits générée aléatoirement et de même longueur que les individus appelée *masque*. Les gènes des individus initiaux sont échangés en fonction de cette chaîne aléatoire, lorsque le bit correspondant vaut 1 [103]. Le croisement uniforme de permutations a les avantages de la simplicité et selon son auteur L. Davis [101, 103], d'une bonne efficacité.

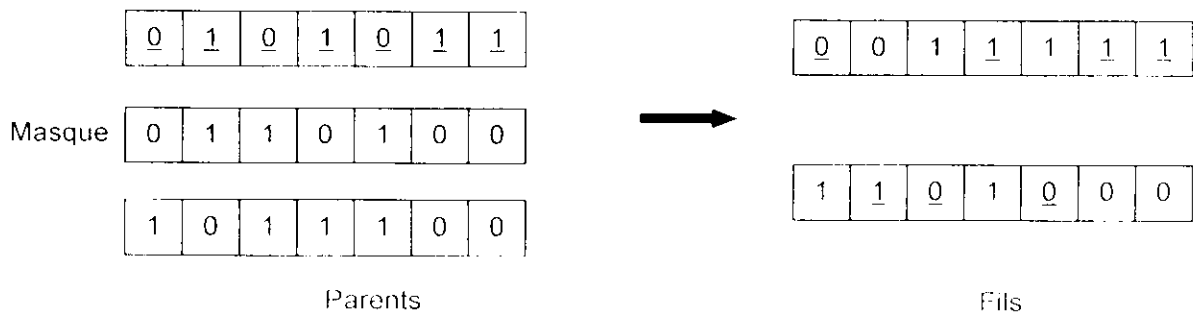


Figure 5 : Exemple du croisement uniforme

Il n'y a pas de critères de choix évidents entre les différentes variantes vues ci-dessus. Pour tel type de problèmes, le croisement « 1-point » sera décevant alors qu'un croisement uniforme fonctionnera mieux. Pour un autre type, la situation sera inverse. L'opérateur de croisement universel capable des meilleures performances pour tous les problèmes n'est pas encore découvert, et il est probable qu'il n'existe pas. Toutefois, en pratique, lorsque le codage binaire est satisfaisant, il apparaît que le choix de la bonne variante est rarement critique, et que les performances varient généralement peu selon l'opérateur choisi.

Un croisement est dit *léta* s'il produit un ou deux individus ayant une trop faible adaptation pour se reproduire. Une solution pour éviter une trop forte proportion de croisement des individus létaux consiste à accoupler préférentiellement les individus qui se ressemblent au niveau génotypique. Toutefois, lorsque le problème est complexe, bien qu'on ait l'assurance que les génotypes croisés ressemblent aux génotypes des parents, cela ne signifie pas que les descendants aient des performances proches de celles des parents. De plus, cette option réduit notablement l'exploration de l'espace de recherche et il peut s'ensuivre des convergences prématurées. Enfin, croiser des individus trop proches les uns des autres ne produit que rarement de la nouveauté, la convergence de l'algorithme s'en trouve ralentie. La restriction de l'appariement est donc une opération délicate à mettre en œuvre.

3.2.4.3 Mutation [11, 12, 13, 14, 100, 101]

Cet opérateur agit sur un individu. Il consiste à choisir d'une manière aléatoire un ou plusieurs gènes (bits) et à modifier leurs valeurs. Chaque bit d'un individu a une probabilité pour qu'il subisse une mutation, notée P_m . La mutation se fait comme suit :

Pour chaque bit de l'individu
Générer un réel aléatoire r , $r \in [0, 1]$
Si $r < P_m$ alors le bit sera inversé.

Plusieurs arguments justifient l'utilisation de la mutation. Ainsi, les actions combinées de la pression de sélection et du croisement peuvent faire disparaître des gènes, aux mêmes positions des génotypes pour tous les individus. La mutation est capable de réintroduire ces gènes dans la population et participe ainsi au maintien de la diversité, utile à une bonne exploration de l'espace de recherche. Elle apporte aussi aux AGs la propriété d'ergodicité de parcours d'espace. Cette propriété indique que l'AG sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution.

D'autre part, elle assure une recherche locale aléatoire autour de chaque individu. Dans cet ordre d'idée, elle peut améliorer considérablement la qualité des solutions découvertes. En effet, le croisement perd de son importance lorsqu'une grande partie de la population est localisée dans les voisinages des maxima de la fonction d'adaptation. Dans ce cas, les individus situés sur un même pic sont souvent identiques et ne subissent aucune modification, ou alors, si les parents appartiennent à des pics différents, les descendants montreront généralement de faibles performances. En revanche, la recherche aléatoire due aux mutations

donne une chance à chaque individu de s'approcher des positions exactes des maxima, autant que le permet la précision du codage.

3.2.5 La nouvelle population

La nouvelle population est obtenue en remplaçant les plus mauvais individus de la population courante par leurs fils, reproduits par la phase précédente, si ses fils sont mieux qu'eux. La littérature des AGs [11, 12, 13, 14, 100, 101] offre une collection fort variée de stratégies dont il convient ici d'en citer quelques unes :

1. La nouvelle population est composée uniquement d'enfants. On laisse alors disparaître tous les individus de la population courante. L'inconvénient majeur de cette approche est le risque de perdre le meilleur individu.
2. Les enfants remplacent d'une façon régulière les individus les moins forts de la génération courante.
3. La nouvelle génération est constituée des n meilleurs individus de la population intermédiaire formée d'enfants et de parents.

3.3 Les paramètres d'un AG

La convergence d'un AG dépend fortement de certains paramètres qu'on doit fixer à l'avance, nous présentons ces paramètres et leurs rôles dans le fonctionnement d'un AG.

3.3.1 La taille de la population

Les conditions de convergence changent avec la taille de la population ; lorsque celle-ci est grande, sa diversité augmente ce qui diminue la convergence vers un optimum local, le temps d'exécution de chaque génération augmente, la recherche risque de s'effectuer d'une façon redondante et l'efficacité de l'algorithme est globalement affectée. Cette taille selon le cas se situe entre 25 et 100 individus. Par contre, si la taille de la population est petite, alors la probabilité de s'attarder sur des minima locaux est grande.

3.3.2 Le taux de croisement

Le *taux de croisement* détermine la proportion des individus qui sont croisés parmi ceux qui remplaceront l'ancienne génération. L'opérateur de croisement est appliqué avec une probabilité P_c , et plus cette valeur est grande plus de nouvelles structures (individus) sont introduites dans la nouvelle génération. En effet, si ce taux a été fixé à 1, tout descendant est obtenu par croisement. Mais quand ce taux est élevé, les structures performantes sont trop fréquemment détruites. Par contre, si ce taux est trop bas, la population n'évolue pas assez vite. En général P_c varie entre 0.25 et 0.70.

3.3.3 Le taux de mutation

L'opérateur de mutation est appliqué avec une probabilité P_m ; si ce taux est grand alors la recherche devient purement aléatoire, la population est diversifiée et l'AG perd de son efficacité

Si au contraire ce taux est faible, la population est moins diversifiée et en plus il y a risque de stagnation. Des études empiriques conseillent pour l'obtention de bons résultats une fréquence qui se situe autour d'une mutation tous les 1000 bits.

A noter que ces paramètres dépendent étroitement du type de problème à résoudre.

3.3.4 Le fossé des générations

Ce paramètre contrôle la proportion de la population qui doit être remplacée à chaque génération. Si ce taux est à 1, l'ensemble de la population est remplacé.

Rapidement K. DE JONG [13, 14, 100, 101] a proposé de ne pas remplacer l'ensemble de la population à chaque génération et pour cela, il a introduit la notion d'écart entre les générations, un nombre compris entre 0 et 1 qui indique la proportion de parents qui sont remplacés par leurs descendants.

3.3.5 Taux de diversité

Il représente le degré de diversité des individus dans la population. Il est essentiel pour la population d'être composée d'éléments variés. Si le taux de diversité est bas, cela indique qu'il y a ressemblance entre les éléments de la population. Une telle population limite l'espace de recherche puisqu'on aura tendance à évoluer dans des régions plus que d'autres ce qui ne garantit en rien la convergence.

3.4 Les étapes d'un AG

Après cette prise de connaissance des fondements des AGs, on présente les étapes et l'évolution d'un AG [11, 12, 13, 14]. L'organigramme ci-dessous nous montre le fonctionnement des AGs et indique clairement leurs différents opérateurs de base.

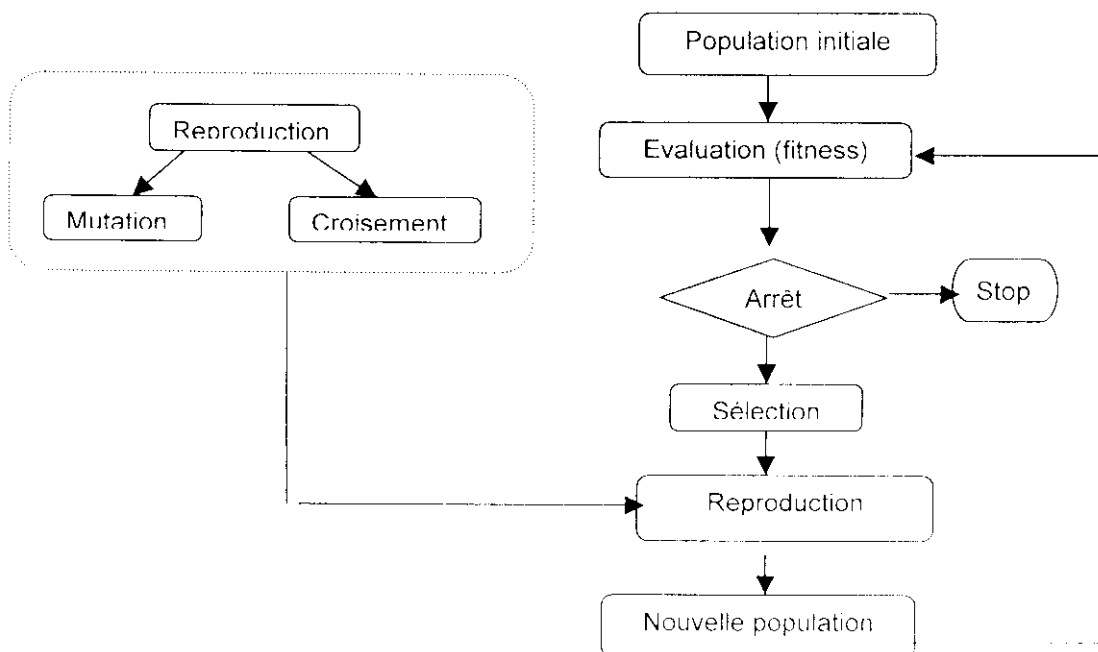


Figure 6 : Organigramme des AGs

L'AG simple accomplit une *génération* lorsque les opérateurs génétiques ont été appliqués simultanément aux individus de la population selon la séquence suivante:

```

DEBUT
T := 0 ;
Initialiser P(T) ;
Evaluer P(T) ;
Tant que (Non Fin)
    T := T+1 ;
    Evaluer P(T) ;
    Sélectionner P(T) parmi P(T-1) ;
Recombinaison P(T) ;
Evaluer P(T) ;
Fin Tant que
FIN
    
```

Les étapes d'évaluation reproduction/ sélection sont appliquées d'une manière itérative sur la population $P(T)$, jusqu'à satisfaction de la condition *Fin* ; cette condition est soit un nombre maximum de générations, soit une convergence vers la meilleure solution.

Au cours de son évaluation, la population tend à converger, c.-à-d., que les individus tendent à se ressembler de plus en plus. Quand la population s'est uniformisée en grande partie, les individus fournissent une bonne approximation d'un optimum du problème. Si cet optimum n'est pas toujours l'optimum global, c'est généralement un optimum local proche (en qualité) de celui-ci.

A noter qu'à chaque exécution d'un AG, les paramètres génétiques sont déjà fixés ; et deux exécutions successives de l'algorithme avec les mêmes paramètres peuvent converger vers des optima différents.

L'application des AGs peut se faire de deux manières différentes :

- Parallèle : on sélectionne deux individus et on leur applique le croisement et la mutation.
- Séquentielle : on sélectionne toute la population (intermédiaire) puis on applique le croisement et la mutation.

Le modèle génétique que nous avons présenté jusqu'à maintenant est l'AG standard décrit originellement par J.Holland [11, 12, 13], mais bien d'autres variantes du modèle génétique existent.

3.5 Les variantes du modèle génétique

3.5.1 L'algorithme génétique canonique

Cet algorithme reste proche de l'algorithme génétique standard AGS sauf que la sélection se fait sur les descendants et leur parents après la reproduction. La reproduction se fait sur des individus issus de la phase dite de clonage (la sélection dans l'AGS). Le schéma de la figure 7 présente bien l'approche d'un algorithme génétique canonique **AGC**.

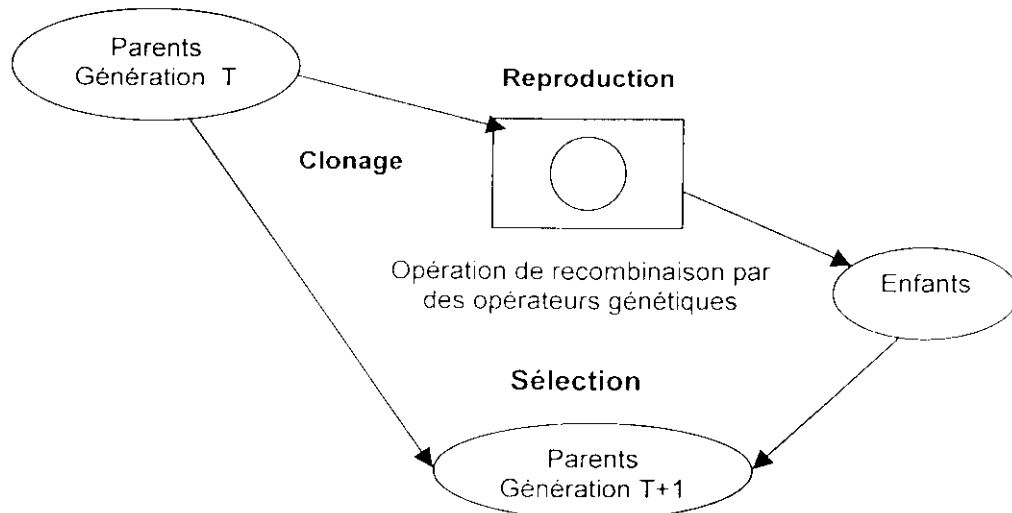


Figure 7 : Principe de l'algorithme canonique

Il s'agit d'une stratégie élitiste puisqu'elle conserve dans la population, d'une génération à l'autre, au moins l'individu ayant la meilleure adaptation. Il apparaît qu'une telle stratégie améliore considérablement les performances de l'AG pour certaines classes de fonctions, mais s'avère décevante pour d'autres classes, en augmentant les taux de convergence prématurés. En fait, une stratégie élitiste favorise l'exploitation des meilleures solutions, se traduisant par une recherche locale accentuée, au détriment de l'exploration de l'espace des génotypes.

3.5.2 L'algorithme génétique modifié

L'algorithme génétique modifié AG-mod est un AGS modifié. La modification par rapport à l'AGS est que dans l'AG-mod, nous n'exécuterons pas l'étape de sélection «sélectionner $P(T)$ parmi $P(T-1)$ », mais plutôt nous sélectionnerons indépendamment r chromosomes (pas nécessairement distincts) pour la reproduction et r chromosomes (distincts) pour mourir. Ces sélections sont faites selon l'aptitude relative des individus : un individu avec une meilleure performance a plus de chance d'être sélectionné pour la reproduction ; les individus avec des performances en dessous de la moyenne ont plus de chance à être sélectionnés pour mourir.

Après l'étape « Sélectionner-parents » et « sélectionner-morts », nous obtenons les trois groupes (pas nécessairement disjoints) d'individus suivants :

- r individus pour la reproduction (parents)
- r individus pour mourir (morts)
- Les individus restants appelés individus neutres.

Le nombre d'individus neutres dans une génération (au moins $N-2r$ et au plus $N-r$ avec N la taille de la population) dépend du nombre de parents distincts choisis et sur le nombre d'individus de chevauchement dans les catégories « parents » et « morts ». Alors une nouvelle population $P(T+1)$ est formée, consistant en $N-r$ individus (tous les individus sauf ceux sélectionnés pour mourir) et r progénitures des r parents.

La structure AG-mod est la suivante :

```

DEBUT
T := 0 ;
Initialiser P(T) ;
Evaluer P(T) ;
Tant que (Non Fin)
    T := T+1 ;
    Sélectionner-parents de P(T-1) ;
    Sélectionner-morts de P(T-1) ;
    Former P(T) ; reproduire les parents ;
    Evaluer P(T) ;
Fin Tant que
FIN

```

Evidemment, la sélection est exécutée de telle manière que les chromosomes les plus forts aient le moins de chance possible de mourir. Une des idées de l'AG-mod est une meilleure utilisation de la ressource disponible d'emmagasinage : la taille de la population

3.6 Association des AGs avec des méthodes locales

La grande force des AGs est leur capacité à trouver la zone de l'espace des solutions contenant l'optimum de la fonction. En revanche, ils sont inefficaces lorsqu'il s'agit de trouver la valeur exacte de l'optimum dans cette zone. Or, c'est précisément ce que les algorithmes locaux d'optimisation réalisent le mieux.

Il est donc naturel de penser à associer un algorithme local à l'AG de façon à trouver la valeur exacte de l'optimum. On peut aisément le faire en appliquant à la fin de l'AG un algorithme local sur le meilleur élément trouvé. Cette technique est d'autant plus efficace que l'on utilise simultanément du clustering (le groupage), et que l'algorithme local est appliqué à chaque meilleur élément de chaque cluster. En effet, on constate souvent que le meilleur élément trouvé par l'AG ne reste pas le meilleur élément après amélioration par l'algorithme local de tous les meilleurs éléments de clusters. Une autre technique consiste à utiliser un algorithme local associé à l'AG pour calculer la fonction objectif d'un élément.

En fait, l'association AG-méthodes locales est une quasi-nécessité. Les deux méthodes sont complémentaires et ont des champs d'application différents. L'AG permet de faire disparaître la combinatoire du problème, laissant alors le champ libre aux méthodes locales dans chacune des zones connexes susceptibles de contenir l'optimum global.

3.7 Le parallélisme

L'intérêt de la parallélisation des AGs est de gagner en temps de calcul. Il existe pour cela au moins deux méthodes classiques utilisées :

- La première consiste à diviser la population de taille N en n sous-populations et à les répartir sur l'ensemble des machines dont on dispose.
- La seconde maintient la population totale sur une seule machine mais les évaluations se font sur les autres machines afin qu'elles se fassent en même temps.

Dans les deux cas, il est nécessaire d'avoir un mécanisme de communication inter-processus.

3.7.1 Parallélisme par îlots

L'idée est de faire fonctionner plusieurs AGs en parallèle avec une population réduite pour chacun. Le programme maître lance N occurrences du programme d'AGs appelées esclaves sur des machines différentes en passant à chacune les paramètres nécessaires à leur bon fonctionnement.

Ensuite chaque processus fait évoluer sa population indépendamment jusqu'à ce qu'il décide (selon une probabilité fixée à l'avance) de rassembler ses meilleurs individus pour en transmettre une certaine quantité (proportionnelle à la taille de la population) à une autre machine de son choix. La machine réceptrice intègre alors ces nouveaux éléments dans sa propre population en éliminant les moins bons de ses individus. L'intérêt du parallélisme par îlots, est qu'il offre la possibilité de travailler sur de grandes populations (N_0) tout en donnant des résultats dans un temps raisonnable puisque la durée nécessaire est à peu de chose près celle qu'il faudrait pour une population de taille $\frac{N_0}{N_m}$, si N_m est le nombre d'ordinateurs disponibles et si l'on néglige le temps de communication.

La méthode introduit un clustering forcé car chaque îlot peut être considéré comme un cluster subdivisé en petits groupes. Chaque machine a la possibilité de converger vers des optima qui seront différents de ceux calculés sur les autres ce qui correspond au comportement introduit avec le clustering. D'autre part, le surcoût de temps passé pour les communications n'est pas excessif puisqu'il n'y a de communication que de temps en temps. Il faut tout de même garder à l'esprit qu'une subdivision en sous-populations de taille trop réduite risque de conduire à faire tourner des AGs non fiables statistiquement. En effet, il faut quand même qu'une population contienne suffisamment d'individus pour que l'espace d'état puisse être exploré de façon correcte, afin que les résultats aient une certaine valeur.

3.7.2 Parallélisation des calculs

Contrairement à la méthode qui vient d'être décrite qui divise la population totale, on utilise ici des démons de calcul de fonctions objectifs dont la seule fonction est de recevoir un individu et de renvoyer son adaptation. Pour utiliser la puissance de calcul parallèle offerte de manière optimale, il faut retarder au maximum les calculs pour les envoyer en blocs aux démons et faire en sorte qu'aucun ne reste inactif. Le principe se résume comme suit : un programme maître se charge de faire la sélection, les croisements, etc. En d'autres termes, il fait évoluer la population, puis répartit les calculs dont il a besoin sur un ensemble de démons. Enfin, dès qu'il reçoit tous les résultats, l'algorithme commence une nouvelle génération.

Il faut noter que ce mécanisme demande un grand nombre de communications pour envoyer les données et les évaluations. La méthode n'est donc intéressante que si le temps

passé pour un calcul d'adaptation est grand devant le temps de communication, elle sera par conséquent utilisée pour des problèmes dont les évaluations de fonctions objectifs prennent du temps, on pense essentiellement à des cas faisant appel à des réseaux de neurones ou de gros calculs matriciels.

La parallélisation est extrêmement efficace pour accélérer les temps de résolution des AGs. Il faut certes bien étudier le problème afin d'utiliser le bon mécanisme de parallélisme

3.8 Applications des Algorithmes Génétiques en Vision

Les possibilités espérées des mécanismes d'évolution ont conduit quelques chercheurs des années 1950 à vouloir les simuler pour les appliquer à l'ingénierie. Mais ces travaux n'ont pas été probants en raison des connaissances insuffisantes, à l'époque, de la génétique naturelle et aussi en raison des faibles performances des calculateurs alors disponibles. Mais depuis le début des années 60, la quantité de travaux sur les algorithmes évolutionnaires suit une croissance exponentielle.

Nous trouvons les AGs dans la finance, l'apprentissage, l'identification, la reconnaissance et les problèmes d'optimisation combinatoire dont un exemple typique : les "problèmes de permutation" (*order based problems*) ; qui consistent à découvrir un ordre des éléments d'une liste maximisant ou minimisant un critère donné (le *problème du voyageur de commerce* [11] en est un exemple). Dans ce qui suit, nous présentons des travaux effectués en vision et en traitement d'images.

- Citons les travaux de S. Bhandarkar [105, 106] qui considère le problème de la détection de contours comme un problème d'optimisation d'une fonction objective de coût en utilisant les AGs. Cette fonction initialement proposée par Tan et al [74, 108] utilise les informations sur les contours et les valeurs des niveaux de gris des régions.

En effet, ils ont défini le problème de la détection de contours comme étant un problème de minimisation de coût. Dans leur travail, ils essayaient de surmonter les imperfections des techniques existantes de détection de contours en formulant une définition d'un contour qui est assez générale pour inclure la plupart des types de contours. Leur travail permet également d'améliorer les techniques existantes de détection de contours en considérant explicitement la structure locale du contour avec ses pixels voisins. Ils définissent le contour comme la frontière dans une image qui sépare deux régions qui ont des caractéristiques sensiblement différentes. La cause de la dissimilitude peut être due à différents facteurs tels que la géométrie de l'objet, la réflectivité extérieure, ou encore à l'illumination. Cette définition du contour inclut une grande variété de types de contours. Par la minimisation de la fonction coût, ils exigent également que les contours détectés soient fins, continus, longs et d'une très bonne localisation.

Tan et al ont implémenté une approche comparative de fonction de coût pour la détection de contours. L'image de contours est obtenue par le calcul du coût qui est fonction de la dissemblance (ou dissimilitude) de la région, de la continuité, de la fragmentation et de la

largeur des structures locales de contours. L'algorithme proposé est basé sur une technique simple de recherche locale [74, 108], il compare localement deux configurations de contours qui sont différentes uniquement en un point, et de sélectionner la meilleure. Les facteurs de coût sont calculés par recherche en arbre de décision.

Le coût ponctuel d'une image binaire de contours S à la position $p=(i,j)$ est obtenu par la somme pondérée des facteurs de coût C_i :

$$F(S,p) = \sum_i W_i C_i(S,p) \quad (4)$$

Le coût total $F(S)$ de l'image entière est obtenu en sommant tous les coûts ponctuels des points de l'image. Les cinq termes de pénalité de la fonction coût sont :

C_d : la dissemblance entre les régions que séparent les contours,

C_t : l'épaisseur du contour,

C_c : la courbure du contour,

C_f : la fragmentation du contour,

C_e : le nombre de points de pixels détectés comme points contours.

Ces facteurs sont destinés à favoriser la formation de contours entre régions de niveaux de gris moyens différents et à pénaliser les configurations dans lesquelles le nombre de points de contours est élevé, et à défavoriser les contours dont l'épaisseur, la courbure, ou encore la fragmentation sont fortes.

Pour chaque structure du contour, les facteurs du coût sont estimés selon l'arbre suivant :

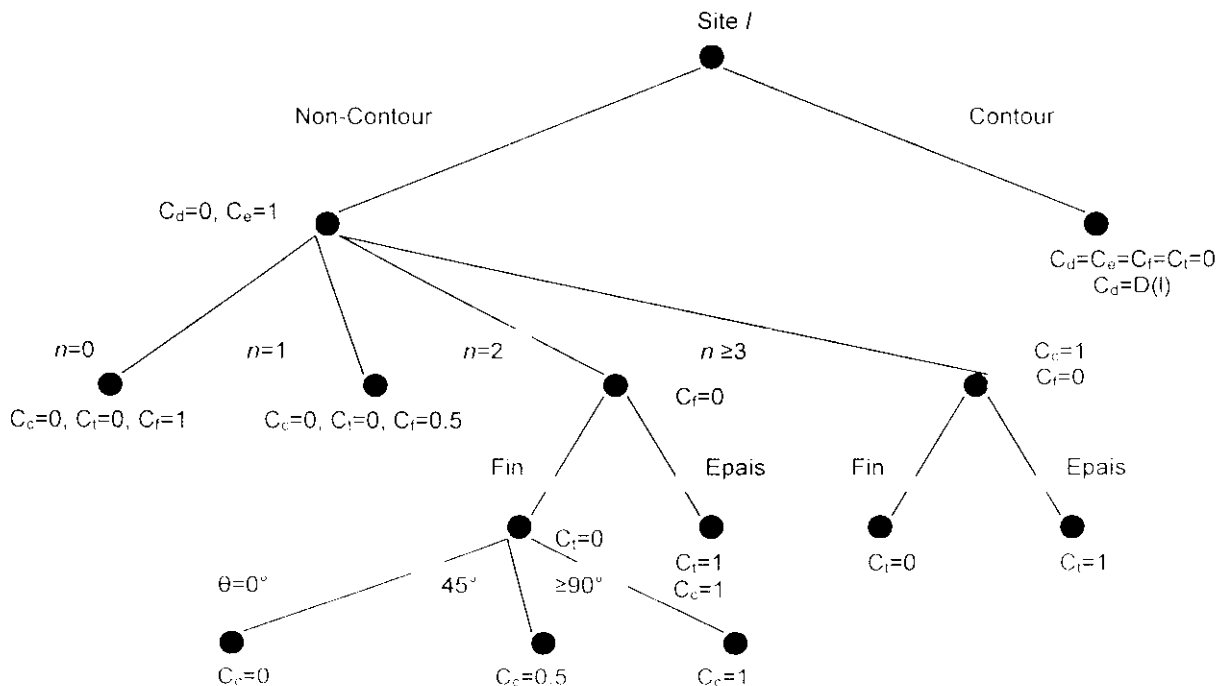


Figure 8. : Décision en arbre pour le calcul des facteurs de la fonction coût

Où : n correspond au nombre de pixels voisins

θ est l'angle associé à la structure du contour

D(l) mesure la dissemblance entre régions séparées par le contour.

La fonction D(l) entre deux régions f(R1,R2) est définie comme étant la différence entre les moyennes de niveaux de gris des deux régions considérées (voir figure 9).

$$f(R_1, R_2) = \frac{1}{n_1} \sum_{p \in R_1} G(p) - \frac{1}{n_2} \sum_{p \in R_2} G(p) \quad (5)$$

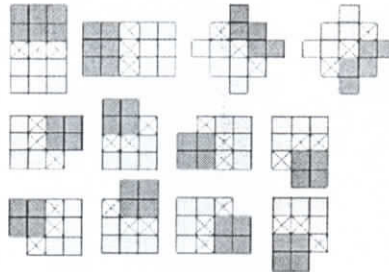


Figure 9: Les régions utilisées pour le calcul de la dissemblance pour les structures de contours valides.

Des exemples de calcul des facteurs de pénalité sont donnés par les figures suivantes :

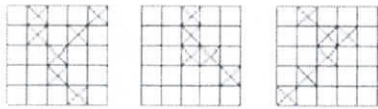


Figure 10: $C_t=0, C_t=1, C_t=1$.

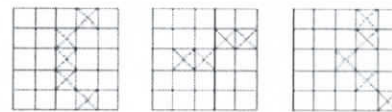


Figure 11: $C_c=0, C_c=0.5, C_c=1$.

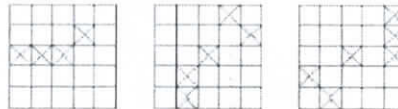


Figure 12: $C_r=0, C_r=0.5, C_r=1$.

Les structures des contours considérés comme étant des contours valides sont définies et données par les figures suivantes:

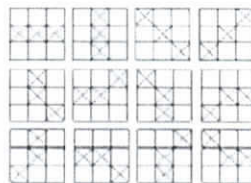


Figure 13: Les structures de contours valides à deux voisins.

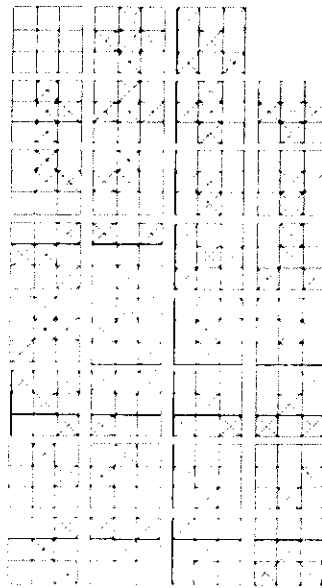


Figure 14 : Les 31 structures secondaires valides

Bhandarkar [105, 106] a exploité les AGs avec cette approche de détection de contours. En effet, avec les AGs, une comparaison est faite entre les structures de contours valides et tous les points contours (considérés sur un voisinage de 3x3) de l'image par utilisation de la fonction fitness basée sur les facteurs de coûts. Ainsi, la recherche de la configuration qui minimise la fonction coût se fait avec les AGs.

Dans ce qui suit, nous citons brièvement d'autres travaux réalisés avec les AGs en vision.

- Bhanu et al [109, 110, 111] ont construit un système de segmentation de scènes d'extérieur en couleur qui optimise, par AG, la segmentation de chacune des images qui lui sont présentées tout en gardant une mémoire de son expérience passée dans une base de règles du type 'condition/ action'. Le système repose sur l'utilisation d'un AG de segmentation paramétrable et sur une fonction d'évaluation de la qualité des segmentations réalisées. La partie 'action' de chacune des règles code sous forme binaire, un jeu de paramètres de l'algorithme de segmentation. La partie 'condition' stocke les caractéristiques d'une image ainsi que l'heure et les conditions climatiques au moment de son acquisition. Lorsqu'une image est présentée au système, elle active les règles porteuses de parties 'condition' correspondant à cette image. Les parties 'action' de ces règles sont alors soumises à un AG dont l'objectif est de générer une paramétrisation conduisant à une segmentation de qualité satisfaisante, au sens de la fonction retenue a priori. Le nouveau jeu de paramètres ainsi obtenu est alors stocké, avec les caractéristiques de l'image, dans une règle de la base du système.
- Dokur et Olmez [113] ont proposé un réseau de neurones quantifié QNN pour la segmentation des images à résonance magnétique MR et des images topographiques CT (Computer Topography). Les éléments du vecteur caractéristique sont formés des intensités

des pixels voisins³ du point considéré. QNN est une structure d'un réseau de neurones conçu par apprentissage avec les AGs. Chaque nœud de la première couche du QNN forme un hyperplan HP dans l'espace d'entrée. Il y a une contrainte sur les HPs dans les réseaux QNN. Le HP est représenté uniquement par un paramètre dans un espace d'entrée à N dimension. Les AGs sont utilisés pour trouver les valeurs optimales des paramètres des nœuds de ces réseaux QNN. Une comparaison pour la segmentation des images MR et des images CT en utilisant le perceptron multicouche et le réseau de kohonen a été donnée. Il a été observé que le QNN présente de meilleures performances de classification en utilisant peu de nœuds avec un temps d'apprentissage court.

Aussi, Matsui et Kosugi [114, 115] proposent une méthode de segmentation des images par les réseaux de neurones classificateurs et la sélection des paramètres est basée sur les AGs. Ces derniers sont utilisés pour la sélection des combinaisons optimales des paramètres d'entrée du réseau de neurones classificateur. L'AG proposé est basé sur l'évaluation d'une fonction objective, qui permet d'estimer la combinaison optimale des paramètres des indices rapidement sans tester les classificateurs actuels. Les deux types de réseaux de neurones utilisés sont : le réseau de neurones par rétro propagation et le réseau de neurones à compteur propagation modifié (modified counter-propagation network).

- Une méthode de segmentation est proposée [116] en utilisant les contours actifs génétiques CAG (genetic snakes), qui sont un modèle actif de contours associé à une procédure de minimisation de l'énergie basée sur les AGs. Les CAG sont proposés afin de remédier à certaines limites des contours actifs classiques, telles que : l'initialisation, l'existence des multiples minima et la sélection des paramètres d'élasticité. Ballerini [116] a prolongé la formulation des CAG de deux manières, en explorant le terme interne additionnel et le terme correspondant à l'énergie externe, il a présenté des applications sur des images couleurs.

Aussi, Xu et al [117] ont utilisé les AGs afin de trouver l'ensemble des paramètres optimaux pour un modèle de contours actifs (*snakes MCA*) qui segmente des lésions dans des images de mammographies. Ces paramètres sont typiquement estimés empiriquement et sont utilisés dans la fonction énergie qui est minimisée par l'algorithme MCA pour réaliser la détection des contours. En utilisant des détections de contours manuelles fournies par des radiologistes, les AGs sont utilisés pour faire varier les valeurs des paramètres jusqu'à ce que les contours produits par l'algorithme MCA soient très proches de ceux donnés par les radiologistes.

- Jiang et Fan [118, 119] introduisent un AG parallèle pour l'analyse des images médicales 3D, qui est un modèle de segmentation et d'enregistrement multimodal d'images. Les résultats expérimentaux montrent que la méthode proposée est encourageante et prometteuse.

³ Chaque point (i,j) est considéré avec son voisinage de taille 3x3 centré en (i,j)

- Alimi [121] propose un système de reconnaissance de l'écriture cursive manuscrite en ligne (on-line) de la langue arabe, et qui peut aussi être adapté aux autres langues. Les mots écrits sont modélisés par la théorie neurophysiologique pour la prédiction de la génération du mouvement, tel que les principaux paramètres extraits de chaque caractère soient des paramètres de l'équation qui décrit la vitesse curviligne de l'écriture. L'AG est utilisé pour la sélection de la meilleure combinaison des caractères à reconnaître par un système beta neuro-flou hiérarchique [121].
- *L'identification d'objets* est une tâche qui peut être confiée à un AG. Le problème consiste à identifier les objets présents sur une image fortement bruitée à partir d'une base d'objets prototypes. Cela pourrait être, par exemple, l'identification d'avions sur des images RADAR. L'AG a pour tâche de découvrir la meilleure position, orientation, mise à l'échelle des objets prototypes, pour sélectionner ceux qui ressemblent le plus possible aux objets présents sur l'image analysée. Le génotype de chaque individu de la population pourrait comporter le numéro d'un prototype, sa position dans l'image, son orientation et son échelle.
- Wada et Imai [122] présentent une segmentation structurale des images de scènes couleurs en utilisant les AGs pour fusionner les limites des régions. La segmentation de l'image est d'abord obtenue avec un algorithme de division-fusion, ensuite, un AG est utilisé afin d'optimiser l'ambiguïté de segmentation dans les limites des régions.
- Une méthode automatique d'extraction des caractéristiques faciales est présentée [123]. Dans l'étape de prétraitement, le visage est approximé à une ellipse, et l'AG est appliqué pour la recherche de la meilleure région elliptique. Ensuite, un AG est appliqué pour extraire les caractéristiques faciales, telles que : les yeux, le nez et la bouche, dans les sub-régions prédéfinies. Des résultats de simulation valident la méthode proposée qui est capable d'extraire automatiquement les paramètres faciaux de différentes images vidéos prises dans un environnement de lumières naturelles en présence d'une certaine quantité de bruit avec des multi faces orientées suivant différentes directions d'angles.
- Takagi et al [124] améliorent l'efficacité du codage des images animées en proposant une méthode de segmentation utilisant les AGs. Dans le codage prédictif du mouvement compensé, l'évaluation de la segmentation de régions et l'évaluation du mouvement montrent qu'ils sont étroitement liés et ils devraient être exécutés de sorte que la complexité du codage et le rapport Signal/ Bruit soient optimisés. Ceci peut être reformulé comme un problème d'optimisation de l'affectation des vecteurs de mouvement à chaque bloc de l'image. Ce papier [124] examine le type de données et propose une méthode de traitement par les AGs telle qu'une segmentation en régions optimale est obtenue. Un résultat de simulation utilisant une image test est donnée.

Aussi, Maslov et al [125] utilise un algorithme évolutionnaire pour le codage de l'image Afin de réduire le temps d'exécution nécessaire à l'évaluation de la fitness du vecteur des

paramètres durant chaque itération, ils proposent d'effectuer certains prétraitements sur l'image avant de l'enregistrer, ainsi, ils proposent d'effectuer un codage en utilisant les fractals ou l'utilisation d'une simple technique de segmentation.

- Un autre travail [126] présente un nouvel algorithme coopératif basé sur l'intégration de l'appariement stéréo et la segmentation. L'appariement stéréo est effectué à partir de deux images stéréo avec l'aide des résultats de la segmentation. En utilisant les AGs en segmentation, nous pouvons raffiner plus efficacement la carte en profondeur. Des résultats expérimentaux sont présentés pour illustrer les performances de la méthode proposée.
- Des procédures adaptatives évolutionnaires pour la segmentation des images couleurs sont décrites dans [127]. Le problème de segmentation est formulé comme un problème d'optimisation et la stratégie évolutionnaire adoptée est basée sur les AGs pour le groupement des petites régions dans l'espace des paramètres couleur. Cette approche utilise les méthodes non supervisées K-Means pour orienter les AGs dans la recherche de la partition optimale ou sub-optimale.
- Les AGs [128] sont introduits pour compenser certaines propriétés indésirables héritées par l'utilisation des algorithmes de la logique floue. L'utilisation de l'AG permet au système de générer automatiquement les fonctions d'appartenance qui sont nécessaires pour modéliser l'incertitude associée aux distributions. Cette approche hybride de la logique floue et des AGs est appliquée pour résoudre un problème typique en traitement d'images : la segmentation. Les résultats obtenus sont meilleurs qu'avec l'algorithme de segmentation par la logique floue.
- Nous trouvons aussi l'utilisation des AGs dans l'industrie. En effet, ils sont exploités pour l'inspection automatique des défauts extérieurs dans des surfaces métalliques [129] en utilisant la vision par ordinateur. Un système expérimental a été développé pour acquérir les images des surfaces métalliques externes et une approche intelligente basée sur la morphologie mathématique et les AGs a été proposée pour détecter des défauts structuraux sur les surfaces métalliques inégales. L'approche emploie l'AG afin d'automatiser l'apprentissage des paramètres du traitement morphologique, tels que l'élément structurant et le seuil de segmentation.
- Il y a aussi l'approche développée par G. Roth et améliorée par E. Lutton [130, 131, 132] qui est fondée sur le codage des primitives existantes dans l'image par un certain nombre de ses points contours et permettant de définir ces primitives (segment, cercle, ellipse, rectangle, quadrilatère), les concaténations des coordonnées de ces points représentent les chromosomes sur lesquels s'effectuent les opérations génétiques.
- Dans l'approche développée par Chun et Yang [133], se ne sont pas des configurations de contours qui évoluent mais des configurations de régions. Une partition initiale en un grand nombre de régions de taille éventuellement réduite est initialement obtenue par un algorithme

de clustering. Un AG est ensuite utilisé pour améliorer cette segmentation en simulant des opérations itérées de fusion et de division des régions obtenues. Un chromosome porte autant de gènes qu'il y a de régions initiales et les allèles sont les labels affectés aux régions dans la segmentation pour laquelle il code. Un chromosome dans lequel des allèles des gène i et j sont identiques spécifie une fusion des deux régions. Une mutation qui modifie l'un de ces deux allèles génère une séparation des deux régions.

- Les AGs sont aussi exploités dans le problème de fusion de plusieurs résultats de segmentation [134, 135], ou encore la reconnaissance et plusieurs autres applications dans les domaines de traitement d'images et de vision (voir [136] jusqu'à [156]), et la liste ne cesse d'augmenter.

3.9 Conclusion

Les algorithmes génétiques sont des algorithmes d'exploration robustes. Lorsque leurs paramètres (taille de la population, nombre d'itérations, probabilités de tirages,...) sont choisis d'une manière adéquate, ils convergent rapidement vers une solution satisfaisante.

Le choix des paramètres d'un AG, ainsi que le codage des données est encore plus un art qu'une science. Il n'y a aucun jeu de paramètres qui serait universel pour tous les problèmes considérés du fait que ces valeurs dépendent étroitement du type de problème à résoudre. Un inconvénient est que pour le moment, on ne sait pas prédire qu'un AG convergera vers la bonne solution

Les AGs ont fait leurs preuves concernant la résolution de certains problèmes d'optimisation complexes. De grandes classes d'applications sont envisagées à ce jour par utilisation des AGs. Nous avons présenté différents travaux effectués dans des applications de la vision et de la segmentation.

Dans le chapitre suivant, nous présentons en détails comment un AG peut être utilisé en segmentation d'images. Nous associons aussi les AGs à la segmentation multifractale en régions homogènes pour assurer une meilleure qualité de segmentation.

Chapitre 4

Applications des Algorithmes Génétiques en Segmentation et Classification des Images

4.1 Introduction

4.2 Segmentation avec un algorithme génétique : La relaxation sélectionniste

4.3 Segmentations Multifractale-Génétique

4.4 Algorithme de relaxation multifractal sélectionniste

4.5 Classification des cellules biologiques du col utérin

4.6 Conclusion

Applications des Algorithmes Génétiques en Segmentation et Classification des Images Cytologiques

4.1 Introduction

Les algorithmes génétiques ont des propriétés qui en font des candidats de choix pour résoudre des problèmes d'optimisation dans lesquels la taille de l'espace de recherche est importante, où les paramètres interagissent de manière complexe et où très peu d'informations sur la fonction à optimiser sont disponibles. Ces propriétés ont motivé leur application à la segmentation d'images.

Nous étudions un algorithme non supervisé de segmentation des images à plusieurs niveaux de gris utilisant les AGs qui nous fournit pour une même exécution l'image de contours et l'image de régions. Les segmentations effectuées sont basées uniquement sur la théorie des AGs, ce qui permettra de montrer l'utilisation et le comportement d'un AG dans la segmentation. Cette étude est principalement basée sur les travaux de P. Andrey [15, 16, 17]. Nous apportons des améliorations à cet algorithme en proposant un nouvel opérateur de mutation qui s'adapte aux caractéristiques spatiales des points et pour assurer la convergence vers l'optimum, le taux de mutation décroît en fonction de la génération.

Vu l'intérêt de l'analyse multifractale dans la segmentation des cellules biologiques (voir chapitre 2), nous proposons avec les AGs d'améliorer la qualité de la segmentation, ce qui permettra de combiner les avantages tout en dépassant les inconvénients des deux théories. Nous proposons ainsi deux algorithmes hybrides : le premier optimise la segmentation multifractale par un AG, le deuxième procède par un étiquetage initial avec les AGs suivi d'une classification finale des pixels en utilisant l'algorithme multifractal non supervisé. Des séries de résultats d'application sur différents types d'images sont données. Les résultats obtenus seront exploités pour classifier les cellules biologiques du col utérin.

4.2 Segmentation avec un algorithme génétique : La relaxation sélectionniste

4.2.1 Description générale de la méthode [15, 16, 17]

La relaxation sélectionniste est une technique générique de segmentation, elle a été appliquée à deux situations, la segmentation par l'intensité lumineuse et par la texture [17]. Dans notre travail, nous nous intéressons à la première situation. Notons que l'algorithme est commun pour les deux situations, ce sont particulièrement les paramètres des vecteurs caractéristiques qui diffèrent.

L'adaptation d'une unité au contexte dans lequel elle se trouve plongée à la position qu'elle occupe sur l'image est l'un des aspects essentiels sur lequel repose le fonctionnement de cette technique. Chaque unité est porteuse d'un chromosome qui la code pour une représentation putative de son environnement immédiat. L'adéquation de cette représentation aux caractéristiques réelles de l'image au voisinage de l'unité est quantifiée par une fonction de fitness. Initialement quelconque, la répartition des unités se réorganise au fil des générations sous l'effet des opérateurs d'un AG distribué à grain fin. En effet, l'AG induit une compétition entre unités. Seules les unités dont les représentations sont les plus pertinentes sont autorisées à se reproduire et à envahir progressivement les régions de l'image. Une même unité ne pouvant posséder une représentation pertinente pour l'ensemble de l'image, la propagation des unités est stoppée aux frontières des régions auxquelles elles sont adaptées. C'est pourquoi la répartition finale des populations d'unités constitue une segmentation de l'image.

4.2.2 Les éléments de la méthode

L'élément de base dans la relaxation sélectionniste est l'unité, à tout moment, il y a une seule à chaque site de l'image.

Une unité est un couple $U=(B, L)$. Le label L représente de manière symbolique l'appartenance de l'unité à une espèce donnée. Le chromosome B est un vecteur de caractéristiques qui constitue une représentation candidate de l'environnement de l'unité. La fitness $f_s(U)$ de l'unité U quantifie le degré de fidélité avec lequel son vecteur de caractéristiques représente les caractéristiques locales de l'image au voisinage du site s qu'elle occupe.

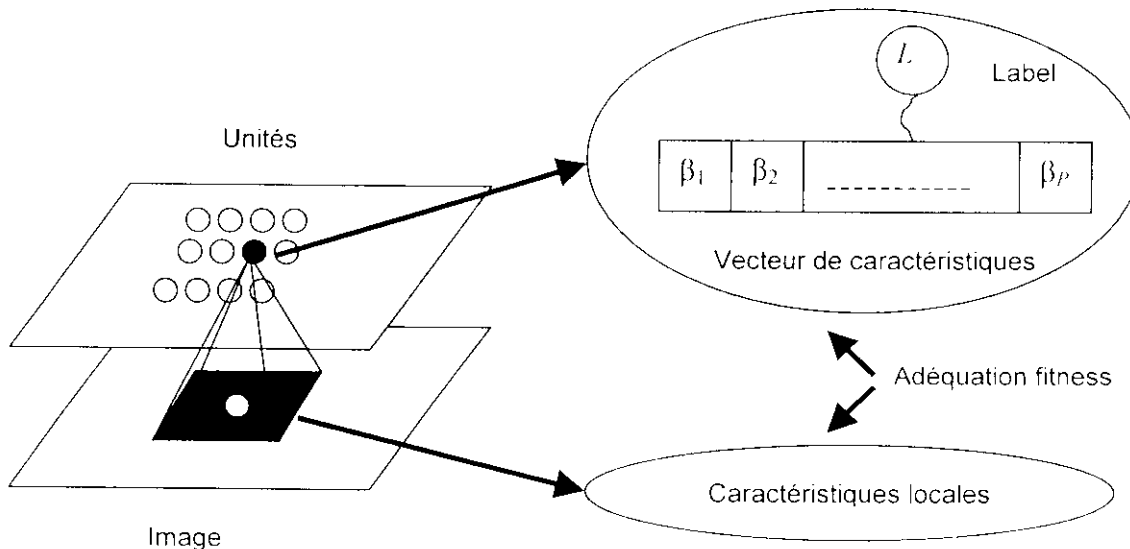


Figure 1 : Structure de la fitness d'une unité $U=(B, L)$

Voisinage : L'évolution de la population d'unité résulte de l'application d'un AG distribué à grain fin. Au cours des étapes de sélection et de recombinaison de chaque itération, les unités interagissent essentiellement avec les unités qui leur sont voisines. A l'exception des unités sur les bords de l'image Le voisinage retenu pour chaque unité est le huit voisins

Unités externes et unités internes: Les unités peuvent être classées en deux catégories distinctes en fonction des unités qui les entourent. **Les unités externes** qui sont à la périphérie d'un groupe composé d'unités porteuses de labels identiques et qui sont au contact d'autres groupes, c.-à-d., qu'ils ont dans leur voisinage au moins une unité dont le label diffère du leur. A l'inverse, **les unités internes** se trouvent plus au centre de ces groupes et ne sont entourées que d'unités porteuses du même label que le leur.

Codage des caractéristiques : Le choix des caractéristiques codées dans les vecteurs portés par les unités est arbitraire et dépend du critère de segmentation choisi a priori. Dans le cas de la segmentation au sens des niveaux de gris, on prendra des matrices de pixels. Dans le cas de la segmentation au sens des textures [17], les paramètres des vecteurs caractéristiques seront des descripteurs de texture. Dans tous les cas, un codage réel des paramètres est adopté, cette représentation est plus appropriée qu'un codage binaire dans la mesure où les vecteurs de caractéristiques sont constitués de valeurs réelles du problème, et elle offre l'avantage d'éviter les barrières de Hamming¹.

Dans le cas d'une segmentation en sens des niveaux de gris, en chacun des sites de l'image, on détermine un vecteur de caractéristiques A en concaténant les niveaux de gris dans la fenêtre $p=k \times k$ centrée sur ce site. La taille k des fenêtres est un paramètre externe spécifié par l'utilisateur et ajustable en fonction de la taille des images à segmenter.

Fonction Fitness : Elle doit quantifier la proximité entre les valeurs de niveaux de gris dans le vecteur porté par une unité et les valeurs des pixels sur l'image à la position qu'elle occupe. La fitness doit être d'autant plus forte que l'écart entre les deux vecteurs est faible. La fonction fitness retenue est l'opposé de la distance de Manhattan entre le vecteur de caractéristiques $B=(\beta_1, \dots, \beta_p)$ de l'unité et le vecteur $A=(\alpha_1, \dots, \alpha_p)$ déterminé sur l'image :

$$f_s(U) = -\sum_{i=1}^p |\alpha_i - \beta_i| \quad (1)$$

s désignant le site occupé par l'unité.

Au cours de l'élaboration de la méthode, l'utilisation de la distance euclidienne dans la fonction fitness a été testée. Comme les segmentations obtenues par l'une ou l'autre de ces deux distances sont comparables, la distance de Manhattan a été retenue parce qu'elle induisait des temps de calculs inférieurs à ceux induits par la distance euclidienne. Notons que le label d'une unité n'intervient pas dans le calcul de sa fitness.

Dans les sections suivantes, nous présentons les étapes génériques de l'algorithme.

¹ Pour qu'un algorithme génétique passe cette barrière de Hamming, il lui faut un grand nombre de mutations simultanées, ce qui est un événement presque impossible

4.2.3 Les étapes de l'algorithme

4.2.3.1 Initialisation

Les labels des unités sont initialement tous distincts. Une unité est dotée d'un label donné par son numéro d'ordre dans un balayage ligne à ligne de l'image. Nous partons donc d'une configuration dans laquelle chaque site est considéré comme une région distincte des autres. En théorie, on peut obtenir des segmentations qui contiennent autant de labels qu'il y a de pixels dans l'image. Les paramètres des vecteurs de caractéristiques des unités sont initialisés par tirage uniforme dans des intervalles dont la définition dépend de l'application considérée.

4.2.3.2 Sélection

Elle permet de sélectionner les individus les plus viables de la population précédente localement au voisinage de chaque site ou individu mais aussi globalement en explorant toute l'image à la recherche des meilleurs individus. Elle est effectuée simultanément en chacun des points de l'image. La population d'unités est donc mise à jour de manière synchrone au cours de l'étape de sélection. Une sélection de type tournoi a été implantée, dont les modalités diffèrent suivant qu'elle est appliquée à un site d'une unité de type *Externe* ou de type *Interne*.

Sélection au niveau d'un site occupé par une unité de type Interne : Un tournoi local est réalisé parmi les unités qui se trouvent dans le voisinage 3x3 de ce site. L'unité choisie pour occuper le site central et remplacer l'unité courante à la génération suivante est celle dont la fitness est la plus forte dans cette population locale. L'unité située sur le site courant n'est donc pas systématiquement remplacée puisqu'elle participe au tournoi.

Sélection au niveau d'un site occupé par une unité de type Externe : Une unité étrangère au voisinage est choisie au hasard et uniformément dans la population toute entière. La fitness de cette unité en chacun des sites du voisinage est calculée et comparée à la fitness de l'unité occupant le site correspondant. Si l'unité étrangère est meilleure que toutes les unités du voisinage, alors elle est choisie pour remplacer l'unité courante. Sinon, un tournoi local est effectué suivant les mêmes règles que celles appliquées au niveau d'un site occupé par une unité Interne. Dans ce cas, c'est donc une unité du voisinage qui remplace l'unité courante.

```
Procédure Sélection
  Pour chaque site  $s$  faire
     $f_{\max} = -\infty$ 
    Pour chaque site  $s' \in \mathcal{N}_s$  faire
      Si  $f_s(U_{s'}) > f_{\max}$  faire
         $f_{\max} = f_s(U_{s'})$ 
         $U^* = U_{s'}$ 
      Fin Si
    Fin Pour
```

```

Si type ( $U_s$ ) = Externe faire
  r = random(S)
  Pour chaque site  $s' \in \eta_s$  faire
    Si  $f_{s'}(U_s) > f_{s'}(U_r)$  faire
      Aller à Etiqu
    Fin Si
  Fin Pour
   $U^* \leftarrow U_r$ 
Fin Si
Etiqu
 $U_s \leftarrow U^*$ 
Fin Pour
Fin Sélection

```

avec :

f_{\max} : la fitness maximale.

S : l'ensemble des sites de l'image.

f_s , U_s et η_s sont respectivement la fitness, l'unité et le voisinage du site s.

4.2.3.3 Recombinaison et Mutation

Ces opérateurs entraînent une dynamique de l'étiquetage initial qui conduit à une configuration stable d'étiquettes et par conséquent à la formation des régions de l'image.

Les unités de type Externe sont exclues de ces mécanismes. D'une part, pour éviter que les contours prennent un aspect très irrégulier à cause des effets éventuellement destructeurs de la recombinaison et de la mutation. D'autre part, cela empêche tout échange de matériel génétique entre unités porteuses de labels distincts. Elles pourraient aussi conduire à la formation de populations d'unités porteuses d'un même label bien qu'adaptées à des régions différentes, ce qui conduirait à une sous-segmentation.

Une telle barrière génétique est nécessaire car des unités dont les labels diffèrent sont censées être adaptées à des caractéristiques, donc à des régions, distinctes de l'image.

Recombinaison : est l'uniforme généralisée. Pour chacune des unités Internes, une unité voisine quelconque est choisie comme partenaire de recombinaison. Chaque allèle de l'unité courante est alors remplacé par un allèle correspondant de l'unité voisine avec une probabilité P_c . La procédure de croisement s'effectue selon le pseudo code suivant, avec :

p : La taille des vecteurs caractéristiques.

$B_{s,i}$: L'allèle du site s à la position i.

```

Procédure Recombinaison
Pour chaque site  $s$  faire
  Si  $\text{type}(U_s) = \text{Interne}$  faire
     $s' = \text{random}(\eta_s)$ 
    Pour chaque  $i \in [1, \dots, p]$  faire
      Si  $\text{random}([0, 1]) < Pc$  faire
         $\beta_{s,i} = \beta_{s',i}$ 
      Fin Si
    Fin Pour
  Fin Si
Fin Pour
Fin Recombinaison
  
```

Mutation : est une mutation ponctuelle. La probabilité qu'un allèle d'une unité Interne soit affecté par une mutation est donnée par le taux de mutation Pm . La nature de la perturbation générée par une mutation dépend du type de segmentation réalisée.

```

Procédure Mutation
Pour chaque site  $s$  faire
  Si  $\text{type}(U_s) = \text{Interne}$  faire
    Pour chaque  $i \in [1, \dots, p]$  faire
      Si  $\text{random}([0, 1]) < Pm$  faire
        Muter ( $\beta_{s,i}$ )
      Fin Si
    Fin Pour
  Fin Si
Fin Mutation
  
```

Plusieurs modes de mutation ont été proposés pour les AGs, Andrey a introduit ce nouveau mode :

L'amplitude de la perturbation subie par un allèle choisi pour être muté est proportionnelle à l'écart type σ_s des niveaux de gris dans le voisinage du site s calculée sur la fenêtre $k \times k$. Ainsi, la mutation du $i^{\text{ème}}$ allèle de l'unité U consiste à mettre à jour la valeur β_i selon :

$$\beta_i \leftarrow \beta_i + \varpi_s \quad (2)$$

où ϖ_s désigne un bruit gaussien centré de variance $m\sigma_s^2$. Où m contrôle l'amplitude de perturbation.

4.2.4 Paramètres de l'algorithme

Plusieurs paramètres doivent être spécifiés par l'utilisateur. Contrairement aux AGs traditionnels, la taille de la population n'est pas un paramètre libre de la méthode, puisqu'elle est automatiquement fixée par la taille de l'image. Les valeurs optimales de la taille des matrices de pixels $p = k \times k$ et le nombre de générations N dépendent de l'image à segmenter

En effet, il peut être nécessaire d'adapter la taille p des matrices de pixels utilisées comme vecteurs de caractéristiques au contenu de l'image à traiter. Il faut prendre plus ou moins

suyant qu'on souhaite lisser ou non les détails fins de l'image. De même, dans la mesure où les unités se déplacent essentiellement de proche en proche, il est nécessaire d'adapter le nombre de générations N à la taille de l'image. Plus une image est grande, plus les régions qu'elle contient sont susceptibles de l'être également, et plus il faudra de temps aux unités pour les envahir.

Les valeurs optimales des paramètres Pc et Pm sont au contraire indépendantes des images à traiter. Pour déterminer ces valeurs optimales, nous pourrions procéder de manière purement empirique et tester un grand nombre de combinaisons différentes des paramètres.

Andrey [15, 16, 17] a effectué une étude basée sur la quantification des segmentations obtenues avec différentes combinaisons des paramètres, en utilisant comme image-test une image synthétique dont la segmentation idéale est parfaitement connue. Les paramètres optimaux pour la recombinaison et la mutation sont ceux qui fournissent les meilleures performances au sens des critères définissant une sur- et sous-segmentation². L'étude effectuée permet d'annoncer les résultats suivants:

- **Mutation optimale :** Sur les N générations qui auront été effectuées au cours d'une relaxation, chacun des allèles de l'une des unités aura subi en moyenne $Pm N$ mutations. Lors de chacune de ces mutations, une valeur tirée de la loi normale $N(0, m\sigma_s^2)$ a été ajoutée à la valeur précédente de l'allèle. Les mutations successives étant indépendantes, leur effet cumulé pour un allèle donné sur l'ensemble des N générations est donc une perturbation qui suit $N(0, Pm.N.m.\sigma_s^2)$. Notons que la mutation au site s dépend de la variance σ_s^2 calculée localement.

Une condition d'optimisation est donnée:

$$m Pm N = 1 \quad (3)$$

Ce qui conduit à prédire que les valeurs optimales pour le taux et l'amplitude de mutation sont inversement proportionnelles. Bien plus, cette condition prédit que les valeurs respectives de Pm et m ne sont pas importantes. En effet, la condition exprime une relation entre l'effet global de la mutation à travers le nombre $m.Pm$ et le nombre de générations N . Pour les valeurs optimales de m et Pm la conclusion suivante est annoncée :

« Le nombre de labels faiblement représentés qui ne correspondent pas à des régions dans l'image est d'autant plus faible que Pm est grand, il est souhaitable de prendre Pm assez grand. Il n'est pas non plus souhaitable de choisir cette valeur au delà d'une certaine valeur limite Pm_l pour laquelle le nombre de labels diminue, puisque cela augmenterait le temps de calcul sans pour autant conduire à des résultats satisfaisants. Dans le cas général, aucun

² Dans une sous-segmentation, il y a au moins un même label qui est attribué à au moins deux régions différentes de la segmentation cible. Pour une sur-segmentation, il y a au moins une région de la segmentation cible à laquelle au moins deux labels distincts ont été attribués.

moyen ne nous permet d'évaluer P_m . Andrey a donc proposé de choisir arbitrairement de fixer $m=1$, ce qui conduit automatiquement à prendre $P_m = \frac{1}{N}$ ».

- **Recombinaison optimale** : Selon l'étude théorique présentée au chapitre précédent, nous énonçons ces remarques :

Pour une dimension donnée de l'espace des caractéristiques, la recombinaison ne modifie en effet pas la distribution des allèles au sein d'une sous-population d'unités. Cet opérateur génétique ne fait que générer des échanges d'allèles, sans en modifier les valeurs.

En plus, plus le taux de recombinaison est élevé, plus le coût moyen du chemin entre les segmentations initiale et finale est faible. Il est plausible que la recombinaison accélère la convergence en permettant l'apparition précoce de combinaisons d'allèles de bonne qualité, grâce aux échanges des produits des premières mutations. En l'absence de recombinaison, ces combinaisons apparaissent également, mais il leur faut plus de temps pour s'établir.

Aussi, dans la mesure où on se rapproche plus d'une bonne segmentation, le nombre de labels est faible et inversement, leur taille est importante, alors le choix de la valeur de P_c qui optimise ces deux quantités est prise égale à 0.5 ; Ce choix est surtout justifié par le fait que la convergence est d'autant plus rapide que le taux de recombinaison est fort.

- Une remarque très importante est à souligner :

« Les valeurs optimales du taux et de l'amplitude de mutation sont indépendantes de la valeur du taux de recombinaison. Réciproquement, la valeur optimale du taux de recombinaison est indépendante des valeurs du taux et de l'amplitude de mutation, tant que leur produit est optimal ».

4.2.5 Variances locales de la mutation

Rappelons que la mutation procède par ajout à l'allèle d'une perturbation tirée dans la loi normale $N(0, m\sigma_s^2)$, avec σ_s^2 est la variance des niveaux de gris de la fenêtre $k \times k$ centrée sur le site s de l'unité à muter. L'application de cet opérateur sur des images de synthèse où les valeurs des pixels sont distribuées suivant des lois normales identiques mais indépendantes les unes des autres présente des résultats satisfaisants. En effet, dans ce type d'images, la variance calculée sur une fenêtre $k \times k$ fournit une estimation satisfaisante de la variance réelle des niveaux de gris de la région dans laquelle elle est localisée.

Cependant, dans le cas des images de scènes réelles, souvent cette estimation n'est pas satisfaisante. En effet, de telles images sont caractérisées par d'importantes autocorrélations entre pixels appartenant à une même fenêtre, en particulier pour les petites valeurs de k . Par conséquent, la variance calculée dans une fenêtre de $k \times k$ pixels en un site s de l'image peut sous-estimer de manière importante la variance réelle de la région à laquelle ce site appartient. Pour corriger cet effet, deux solutions ont été proposées [15, 16, 17].

La première solution consiste en l'estimation de la variance sur une fenêtre de taille plus importante que la taille des matrices de pixels codées dans les unités. Cependant, cette solution présente l'inconvénient d'étendre l'influence spatiale des contours des régions.

La deuxième solution consiste à multiplier la variance calculée sur une fenêtre $k \times k$ par un facteur correcteur. Andrey propose une procédure d'estimation et énonce ces résultats :

« la variance globale est en moyenne approximativement 2 fois plus grande que la moyenne des variances calculées sur des fenêtres 3×3 et 1.5 fois plus grande que la moyenne des variances calculées sur des fenêtres 5×5 . Ces valeurs sont utilisées pour corriger les estimations locales des variances σ_i^2 qui interviennent lors des mutations ».

Il faut noter que ces valeurs sont estimées empiriquement et que les performances de l'algorithme sont d'autant meilleures que la distribution des niveaux de gris des régions à segmenter est compatible avec le modèle stationnaire gaussien.

Et souvent, les régions dans la scène ne sont pas caractérisées par des variations locales et rapides des intensités lumineuses, mais sont au contraire assimilables à des surfaces qui résultent de l'illumination non uniforme de la scène. De telles régions ne satisfont pas le modèle stationnaire gaussien sur lequel repose la mutation et l'analyse de l'algorithme.

Pour pallier à ces inconvénients, il est plus intéressant de choisir un mode de mutation qui s'adapte aux caractéristiques spatiales et d'intensité de chaque pixel. Ainsi, nous proposons d'effectuer la mutation comme suit :

« L'allèle du gène à la position i du chromosome à muter sera remplacé par l'allèle du gène à la position i du chromosome choisit aléatoirement parmi les 8 voisins ».

Ce choix nous a permis d'éviter le calcul des variances au voisinage de chaque chromosome à muter, de rendre l'implémentation de la mutation plus simple et gagner en temps de calcul.

Aussi, pour diminuer l'effet de la mutation afin que la recherche ne devienne pas aléatoire au fur des générations, nous proposons d'appliquer la mutation avec une probabilité qui décroît selon la génération. En effet, des résultats de recherche [130, 131] ont montré que des AGs à population finie et probabilité de croisement constante assurent théoriquement une convergence vers l'optimum global, si la probabilité de mutation P_m suit une loi de décroissance en fonction de la génération k :

$$P_m(k) = \frac{1}{2} * k^{-M/L} \quad (4)$$

M est la taille de la population, L la longueur des chromosomes.

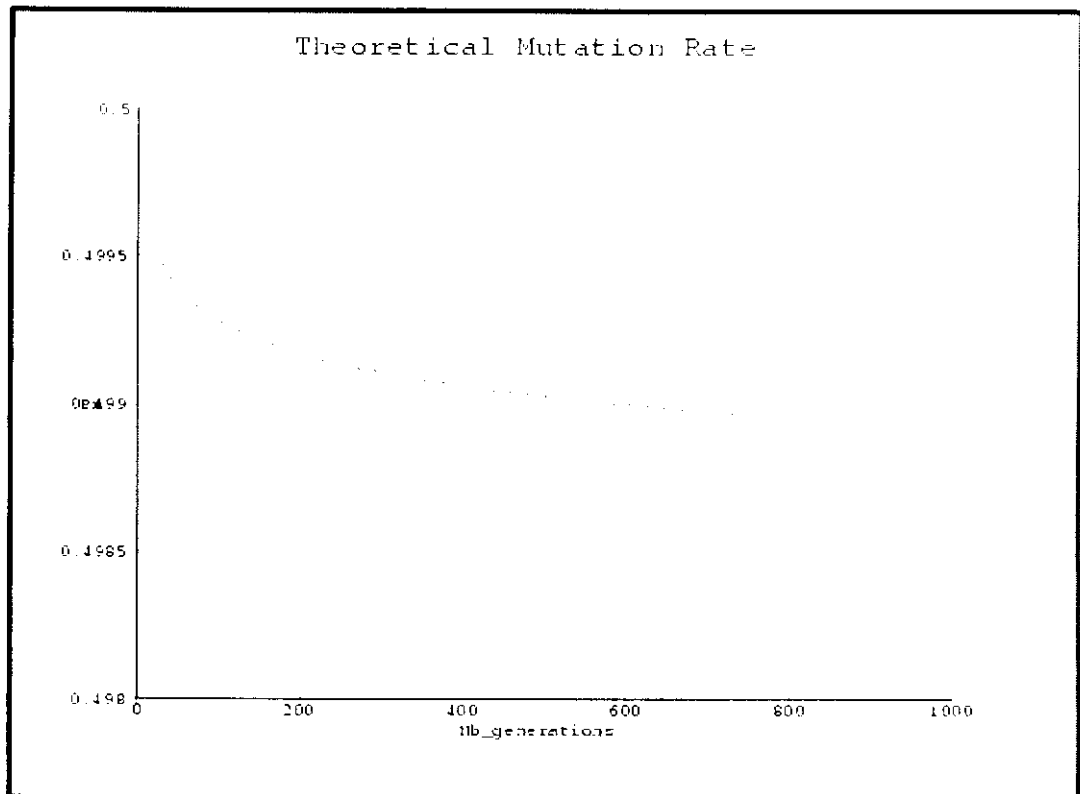
Ce taux de décroissance est très lent, et un nombre infini de générations est nécessaire pour assurer la convergence de l'AG vers l'optimum global. D'un point de vue pratique, il est plus intéressant d'utiliser un taux de décroissance plus rapide. Les auteurs de [130, 131] proposent d'utiliser un taux de décroissance plus rapide donné par la formule suivante

$$Pm_2(k) = Pm(0) * \exp\left(-\frac{k}{\alpha}\right) \quad (5)$$

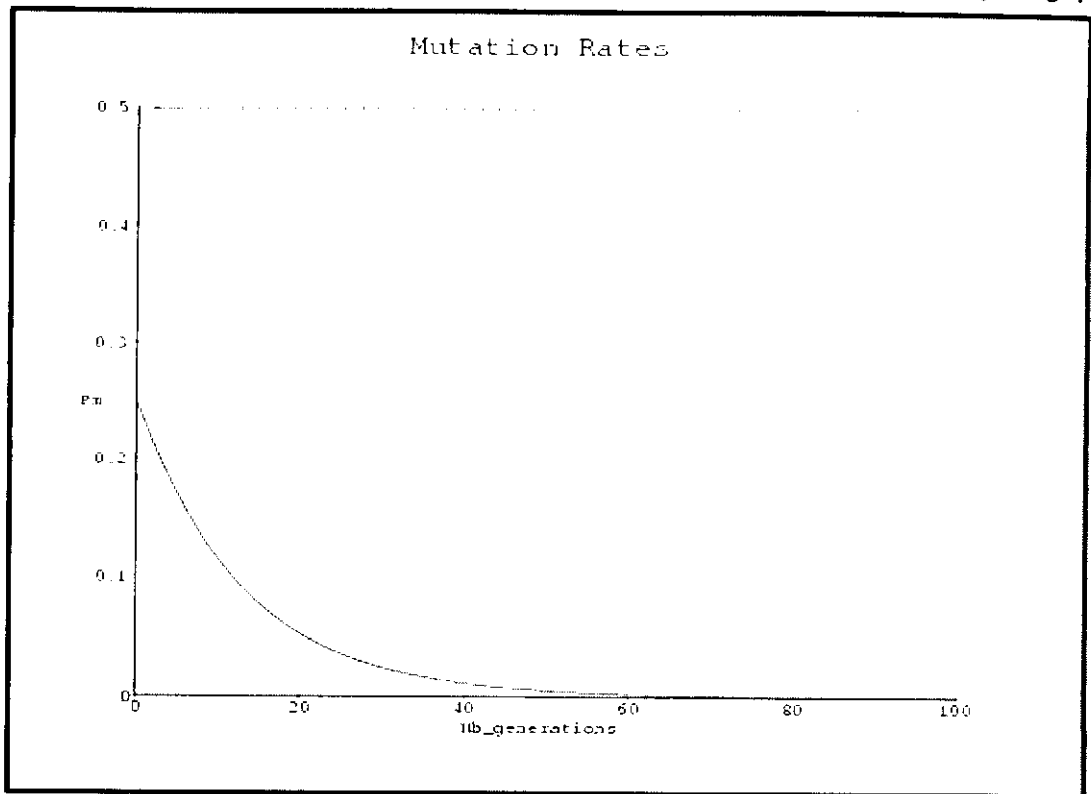
$P_m(0)$ est la probabilité de mutation initiale, α est calculé pour donner un taux final de mutation très bas (exemple de 10^{-4}) :

$$\alpha = \frac{\text{Nbre maximale de générations}}{\ln\left(\frac{Pm(0)}{10^{-4}}\right)} \quad (6)$$

Les courbes de la figure suivante présentent les deux fonction de décroissance, (a) la courbe Pm_1 sur 1000 générations, (b) la fonction Pm_2 sur 100 générations.



(a)



(b)

Figure 2 : Les courbes de décroissances de l'opérateur de mutation.

Dans la section suivante, nous présentons une série de résultats d'application de l'algorithme de relaxation sélectionniste en utilisant notre mode de mutation proposé.

4.2.6 Application à la segmentation des images de scènes réelles

L'algorithme a été appliqué sur différentes images de scènes réelles à 256 niveaux de gris, de format .bmp, de taille 256X256. Ces images n'ont subi aucune transformation ou traitement préalable à leur segmentation par l'algorithme de relaxation sélectionniste.

Les choix des paramètres $p=k \times k$ (taille des matrices), P_c et P_m s'appuient sur l'étude présentée précédemment. Ainsi, pour toutes les images nous choisissons :

- ✓ $p=3 \times 3$,
- ✓ $P_c=0.5$,
- ✓ $P_m(0)=0.03$ et pour chaque génération k : $P_m(k)$ est estimée par (5) et (6).

Les unités se propagent essentiellement de proche en proche, il faut en effet a priori un temps proportionnel à la taille des images pour qu'elles puissent envahir les régions.

Par les figures 3 et 4, nous présentons les contours obtenus avec l'image des dés et l'image d'une cellule avec l'algorithme de relaxation sélectionniste pour différentes valeurs du nombre de générations $N \in \{50, 100, 150\}$.

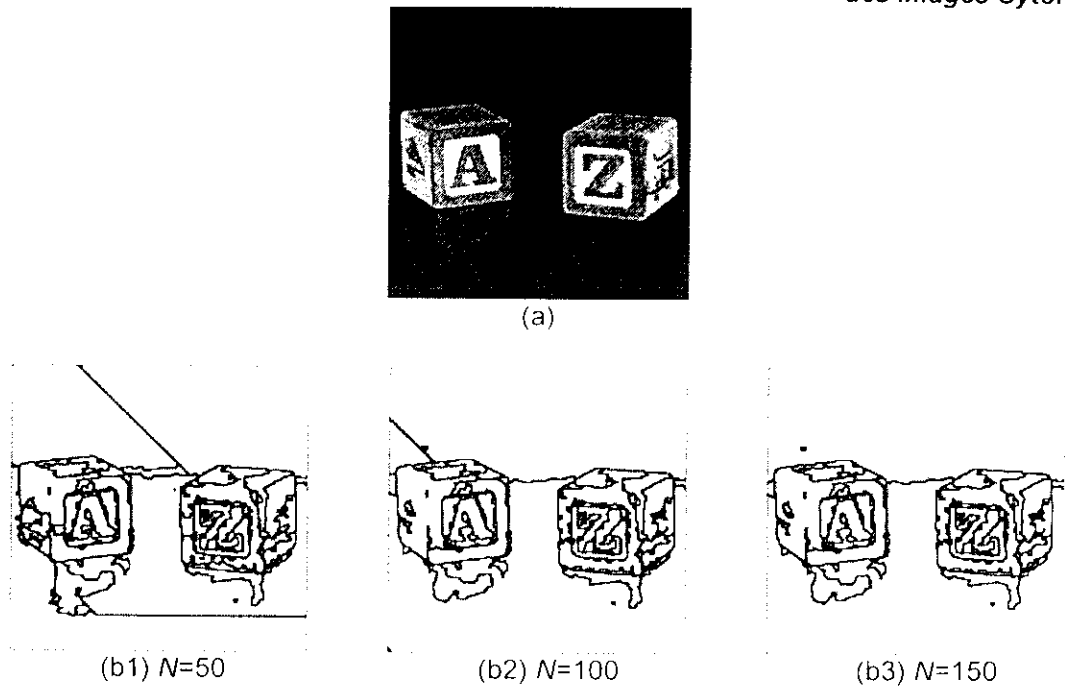


Figure 3 : Détection de contours de l'image des dés avec l'algorithme de relaxation sélectionniste pour différentes valeurs de N .

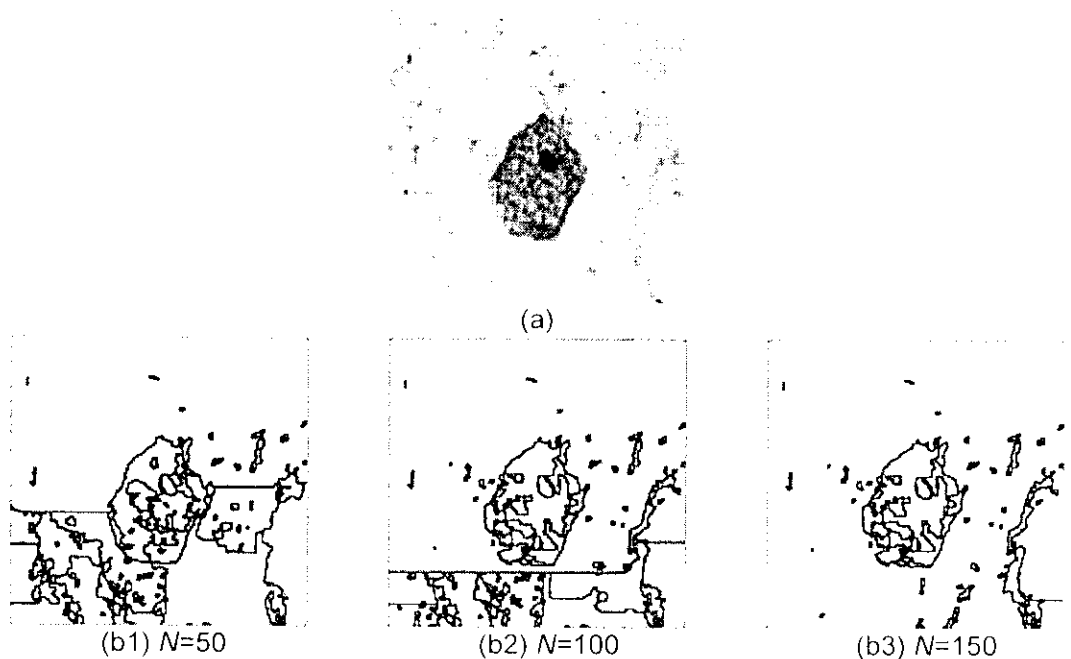


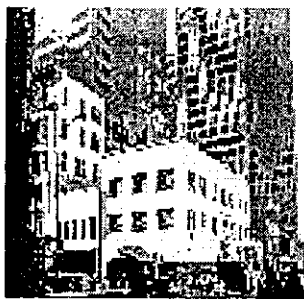
Figure 4 : Détection de contours d'une image de cellule avec l'algorithme de relaxation sélectionniste pour différentes valeurs de N .

Notons que les points contours correspondent aux unités contours qui sont détectées très rapidement dès les premières générations et s'améliorent au cours des générations.

En analysant les images des contours des figures 3 et 4, nous constatons que les étiquettes se stabilisent avec le nombre de générations qui augmente, puisque les contours obtenus convergent vers les limites des régions dont les moyennes des niveaux de gris sont différentes

Pour obtenir les images régions, il suffit d'affecter à tous les points du même label un niveau de gris qui est estimé par la moyenne des intensités des pixels de la région à laquelle le label est affecté. Ceci n'altère en rien la segmentation obtenue et nous permet une visualisation des labels plus parlante puisque les labels n'ont qu'une valeur symbolique.

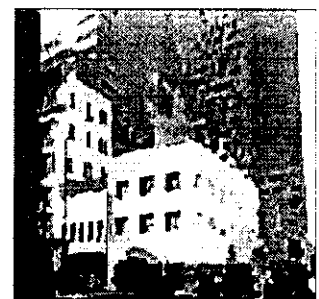
Dans la plupart des cas la valeur $N=150$ est suffisante pour que l'algorithme converge vers une segmentation globalement stable. Cependant, du fait de la variabilité inhérente de la méthode et du contenu des images, ceci n'est pas systématiquement vérifié. Pour cela, nous appliquons notre algorithme pour trois valeurs de N ($N=\{50, 100, 150\}$). La segmentation la plus stable est retenue comme segmentation finale de l'image. Par la figure 5, nous présentons des résultats d'applications sur différents types d'images (objets, scènes réelles, muscles, images cérébrales et images de cellules du col utérin).



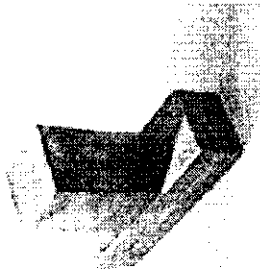
(a1)



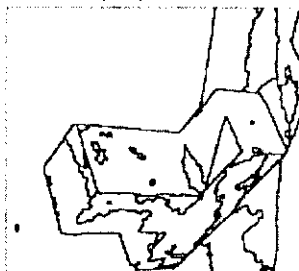
(b1)



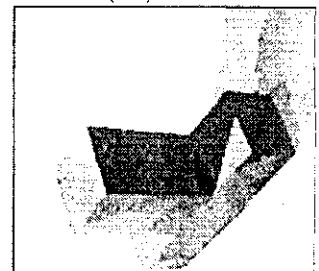
(c1)



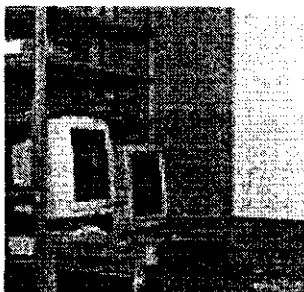
(a2)



(b2)



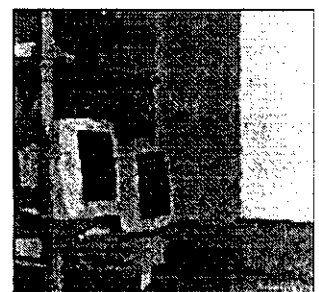
(c2)



(a3)



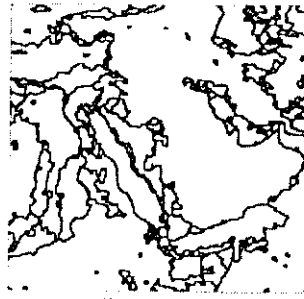
(b3)



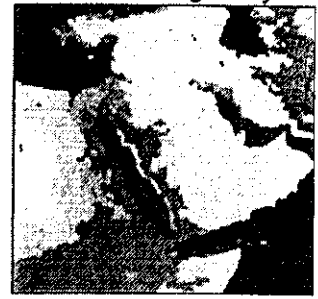
(c3)



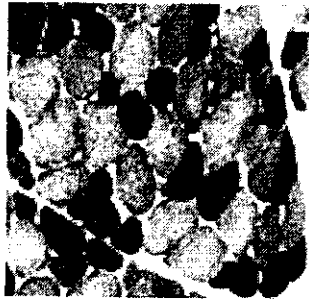
(a4)



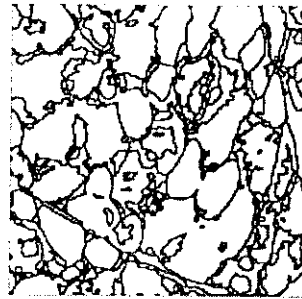
(b4)



(c4)



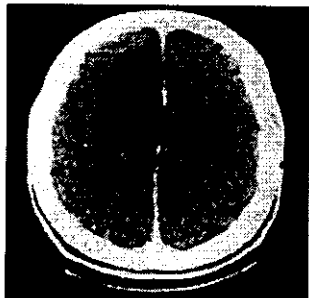
(a5)



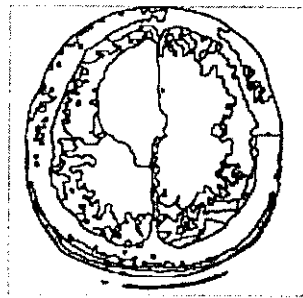
(b5)



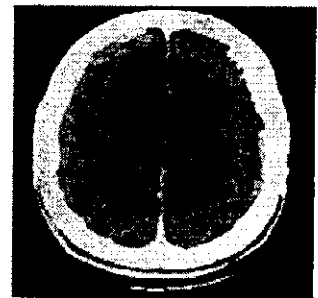
(c5)



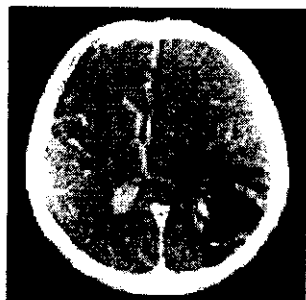
(a6)



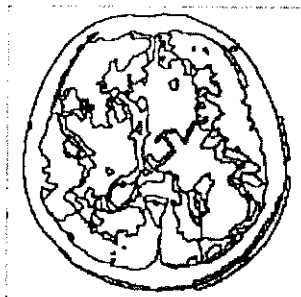
(b6)



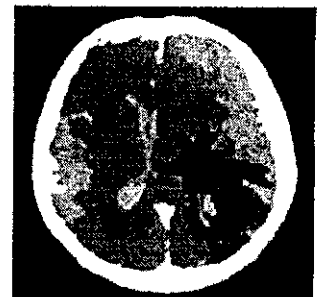
(c6)



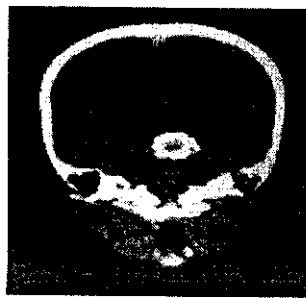
(a7)



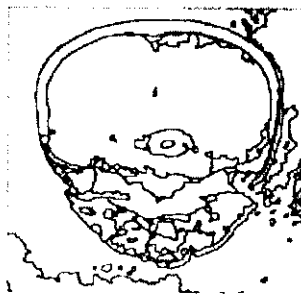
(b7)



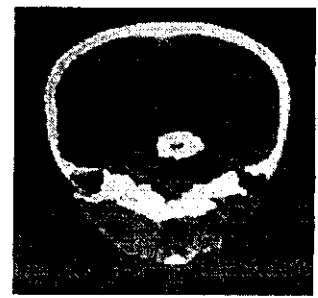
(c7)



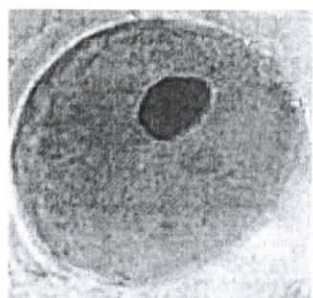
(a8)



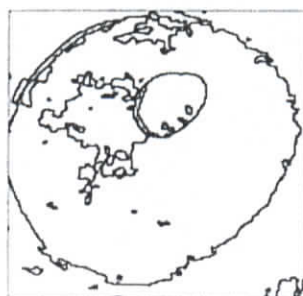
(b8)



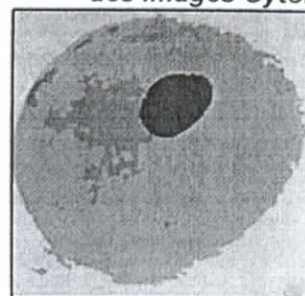
(c8)



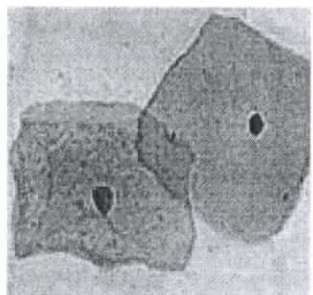
(a9)



(b9)



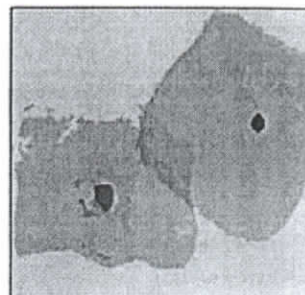
(c9)



(a10)



(b10)



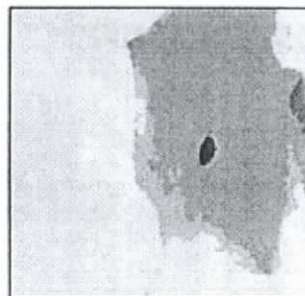
(c10)



(a11)



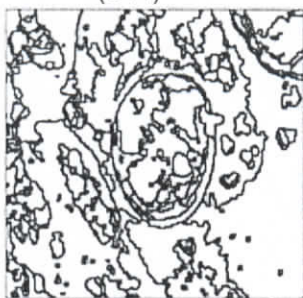
(b11)



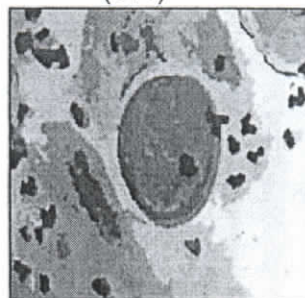
(c11)



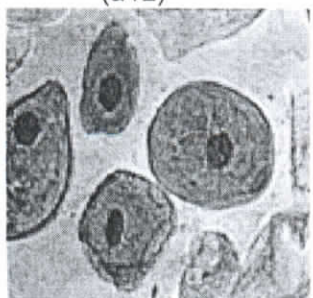
(a12)



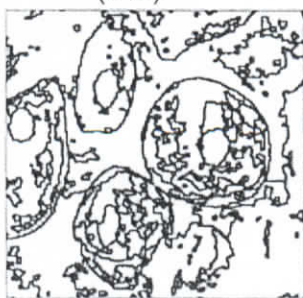
(b12)



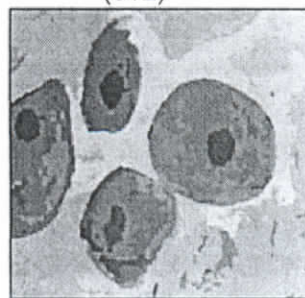
(c12)



(a13)



(b13)



(c13)

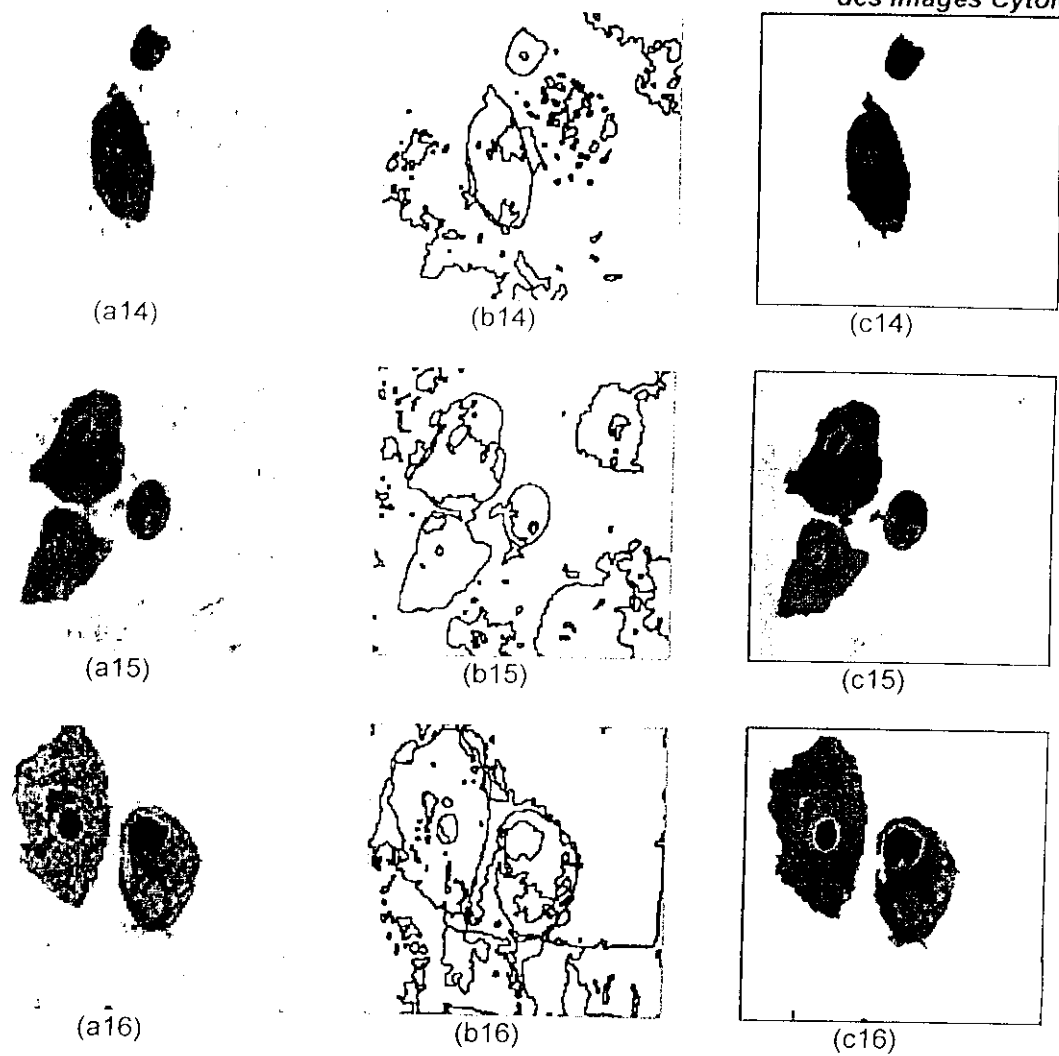


Figure 5 : Segmentation par l'algorithme de relaxation sélectionniste.

Bien qu'on ne dispose pas de critère quantitatif pour évaluer les résultats obtenus, nous considérons que la segmentation obtenue est bonne puisque des labels différents sont attribués à des régions dont les niveaux de gris moyens sont distincts.

L'intérêt de la relaxation sélectionniste est sa capacité de détecter même les régions dont l'intensité moyenne est très proche de celles des régions voisines, c'est l'un des intérêts de cette technique spatialement adaptative. Notons que ces régions qui sont distinctes au sujet humain ne sont souvent pas détectées par la plupart des algorithmes de segmentation. Des exemples de ces régions : les zones non homogènes du fond des images, les hétérogénéités dans les cytoplasmes, les parties communes entre cellules chevauchantes, ...

Notons aussi que même pour les régions où les variations locales des niveaux de gris sont beaucoup plus importantes, la relaxation sélectionniste fournit une segmentation satisfaisante, contrairement à certaines méthodes (telles que les techniques de seuillage).

Sachant que l'algorithme travaille sur des données réelles, la non nécessité des étapes de codage et décodage rend l'algorithme rapide. En plus, l'utilisation de notre opérateur de mutation proposé permet d'assurer une bonne qualité de segmentation et de gagner en temps.

contrairement à l'opérateur de mutation (2) qui nécessite le calcul des variances sur chaque imagette, sachant qu'il y a 65536 chromosomes à traiter à chaque génération pour une image de taille 256x256 et avec le nombre de générations N , le temps d'exécution devient important.

Notons que pour certaines images, la segmentation la plus stable est atteinte pour un nombre de générations N faible (au maximum 50) et l'augmentation de N n'est plus nécessaire.

Rappelons que seule l'information contenue dans les niveaux de gris moyens a été prise en compte pour générer ces segmentations. D'autres caractéristiques des régions, telles que par exemple des informations relatives à la texture peuvent être utilisées.

La caractéristique principale de la relaxation sélectionniste est : « Le nombre de régions et leurs caractéristiques, informations sans lesquelles la segmentation ne peut être réalisée, sont déterminées automatiquement au cours du processus de relaxation ».

Cependant, souvent dans des applications bien spécifiques, des classes ou des objets bien précis doivent être segmentés. Par exemple, pour les images de cellules biologiques, la segmentation doit séparer les noyaux et les cytoplasmes puisqu'ils seront exploités pour la classification et la reconnaissance. Et vu l'intérêt des exposants de singularité dans la séparation des cellules, dans la suite de notre travail, nous proposons d'utiliser les AGs avec les multifractals pour segmenter des images en classes où chacune correspond à une partie bien précise des cellules.

4.3 Segmentations Multifractale-Génétique

D'après la littérature, les AGs sont déjà exploités avec d'autres théories et techniques de segmentation, tel que : ondelettes, snakes, morphologie mathématique, dichotomie d'histogramme, logique floue, ... Dans notre travail, nous proposons de combiner les multifractals avec les AGs pour avoir une plus grande efficacité de segmentation.

Ainsi, nous proposons deux algorithmes : le premier optimise avec les AGs la segmentation obtenue par l'algorithme multifractal, le deuxième procède par un étiquetage initial avec les AGs suivie d'une classification des pixels en utilisant l'algorithme multifractal non supervisé.

Dans ce qui suit, nous présentons ces deux algorithmes hybrides.

4.3.1 Algorithme Multifractal-Génétique

L'objectif est d'essayer d'améliorer la segmentation multifractale en régions homogènes en utilisant les AGs.

Rappelons les avantages de l'algorithme multifractal :

- ✓ un bon moyen de mesure de la texture,
- ✓ une bonne précision dans la détection des limites des régions,
- ✓ et elle n'est pas gourmande en temps de calcul.

Aussi, chaque classe de l'image correspond à un intervalle bien précis de α . Et en exploitant l'algorithme automatique³ de recherche des seuils, nous pouvons avoir les différentes classes principales des images.

Mais pour certaines images très complexes, un même objet de la scène peut être constitué de plusieurs régions non homogènes, d'où confusion entre les différentes classes de l'image et des erreurs de segmentation apparaissent.

Nous proposons de remédier à ce problème en utilisant les AGs. Ainsi, l'algorithme hybride Multifractal-génétique procède comme suit :

- ✓ Estimation des exposants de singularité de tous les points de l'image.
- ✓ Application de l'algorithme de seuillage automatique pour l'estimation des intervalles iso-alpha des différentes classes de l'image.
- ✓ Optimisation de la segmentation en appliquant un AG sur chaque classe correspondant à un intervalle I_{α} .

Dans la section suivante, nous présentons avec plus de détails la dernière étape génétique.

4.3.2 Description de la méthode multifractale-génétique [157]

L'image d'entrée est segmentée en utilisant l'algorithme multifractal non supervisé. L'image segmentée est l'image d'entrée du module d'optimisation par les AGs.

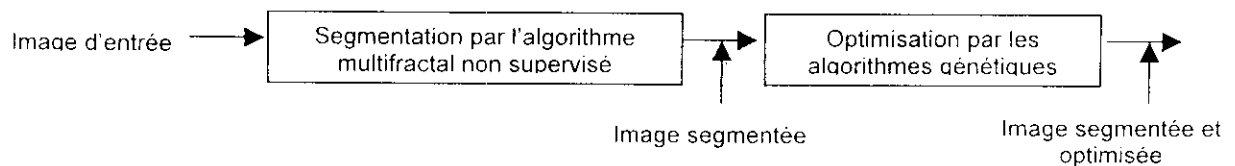


Figure 6 : Schéma de l'algorithme hybride Multifractal-Génétique.

Notre algorithme se base sur ce principe :

Les différentes classes de l'image sont obtenues par l'algorithme multifractal non supervisé, où chacune correspond à un intervalle des exposants de singularité. Pour chaque classe, nous générons aléatoirement des chromosomes de valeurs en exposants de singularités du pixel central et de ses 8 ou 24 voisins. Nous travaillons donc dans l'espace des α . L'objectif est de trouver pour chaque classe, un vecteur qui la caractérise le mieux, en appliquant un AG sur tout ou une partie des chromosomes de la classe.

Donc, pour chaque classe de l'image un vecteur optimal qui la caractérise est estimé. Ensuite, tous ces vecteurs caractéristiques des différentes classes sont utilisés pour obtenir l'image finale de segmentation. Et ceci, en effectuant une classification des pixels par estimation des distances entre le chromosome du pixel à la position s et tous les chromosomes

³ L'algorithme automatique de calcul des seuils sur l'intervalle $[\alpha(\min), \alpha(\max)]$ est appliqué après échantillonnage de l'intervalle $[\alpha(\min), \alpha(\max)]$ avec un pas fixe (par exemple de 0.01)

caractéristiques des classes. Finalement, chaque pixel est affecté à la classe à laquelle il est le plus proche, c.-à-d., dont la distance est la plus petite.

Dans ce qui suit, nous présentons en détails les deux étapes du module génétique à savoir : la modélisation et la classification.

4.3.2.1 Module de modélisation

Ce module conditionne la qualité de la segmentation. Il consiste à extraire pour chaque classe un chromosome de composantes en exposants de singularité de taille 9 ou 25 qui caractérise le mieux la classe. Pour cela, nous exploitons l'AG classique.

Les différentes étapes de cette phase sont les suivantes :

1. Pour chaque classe de l'image définie par son intervalle des exposants de singularité I_c , générer aléatoirement N_p chromosomes qui constituent la population initiale de chaque classe.
2. Sur chacune des populations de chromosomes des différentes classes, un algorithme génétique standard est appliqué afin d'extraire le meilleur chromosome qui modélise le mieux sa classe.

Remarque : Notons que dans cette approche, on se passe d'une propriété très propre aux AGs qui est le codage des individus. Ce qui nous offre un meilleur gain en temps de calcul et évite la barrière de Hamming.

Nous considérons que le chromosome caractéristique de la classe est celui qui est le plus proche du vecteur moyen de la classe. Donc, la fonction fitness utilisée correspond à la distance entre chaque chromosome et le chromosome moyen de sa classe. Donc, il faut minimiser la distance suivante :

$$\min \left(\sum_{j=1}^l | \text{chromosome}[j] - \text{moyenne}(\text{classe}[j]) | \right) \quad (7)$$

Ce qui correspond à maximiser la fonction suivante :

$$- \left(\sum_{j=1}^l | \text{chromosome}[j] - \text{moyenne}(\text{classe}[j]) | \right) \quad (8)$$

Où l représente la longueur du chromosome (taille de la fenêtre du voisinage).

On peut faciliter le calcul de la fitness d'un individu en utilisant un seuil (seuil_f), et le calcul de la fitness devient :

Si la valeur absolue entre le $i^{\text{ème}}$ gène du chromosome et le $i^{\text{ème}}$ gène du vecteur moyen de la classe Cl est inférieure au seuil alors incrémenter la valeur fitness f du chromosome de 1.

$$| \text{Chromosome}[i] - \text{moyenne}[Cl] | \leq \text{seuil} \text{ alors } \text{fit}[i] = 1 \text{ et } f = f + \text{fit}[i] \quad (9)$$

Les unités ayant les valeurs fitness les plus élevées sont sélectionnées, recombinaées et mutées.

Mode de sélection : La sélection se fait en utilisant la méthode de la roue de loterie

Croisement : Un croisement uni-point est utilisé.

Mutation : Muter le $i^{\text{ème}}$ gène du chromosome au site s avec une probabilité Pm consiste à remplacer son allèle par une valeur prise aléatoirement dans l'intervalle $I_{i,s}$ de la classe à laquelle il appartient.

L'AG est appliqué sur chaque classe pour un nombre de générations N .

Une fois que les bons chromosomes des classes sont obtenus, l'algorithme procède à la classification des chromosomes de toute l'image.

4.3.2.2 Procédure de Classification

C'est l'étape finale, elle procède par l'estimation de la distance entre le chromosome de chaque pixel s et les différents vecteurs caractéristiques des classes de l'image.

Ce calcul de la distance s'effectue de la même façon que celui de la fitness de l'étape de modélisation, donc en utilisant l'équation (9) avec un seuil ($seuil_2$). Ainsi, chaque pixel sera affecté à la classe à laquelle il est le plus proche, c.-à-d., ayant une distance minimale.

Nous pouvons utiliser d'autres critères pour effectuer la classification des pixels, telle que la variance pour mesurer la dispersion des chromosomes.

Remarque : Les deux seuils $seuil_1$ et $seuil_2$ ne sont pas nécessairement égaux.

4.3.3 Applications

Par la figure suivante, nous présentons l'application de l'algorithme multifractal-génétique sur les images de cellules biologiques, l'objectif est de séparer les noyaux des cytoplasmes, et ces derniers du fond de l'image. Après l'obtention des trois intervalles iso-alpha des différentes classes avec l'algorithme multifractal non supervisé, l'AG sera appliqué sur chacune des classes. Notons que nous avons adapté les tailles des populations N_p selon celles des classes cellulaires tout en prenant en compte le temps d'exécution. Nous avons ainsi pris ces valeurs :

- noyau ($30 \leq N_p \leq 50$),
- cytoplasme ($150 \leq N_p \leq 500$),
- fond ($500 \leq N_p \leq 1000$).

La première colonne de la figure 7 présente les images originales, la deuxième les résultats de segmentation multifractale et la dernière ceux obtenus après optimisation par les AGs.

Les pixels de la classe 'noyau' (respectivement : cytoplasme et fond) sont présentés en niveau de gris 'noir' (respectivement : gris et blanc).

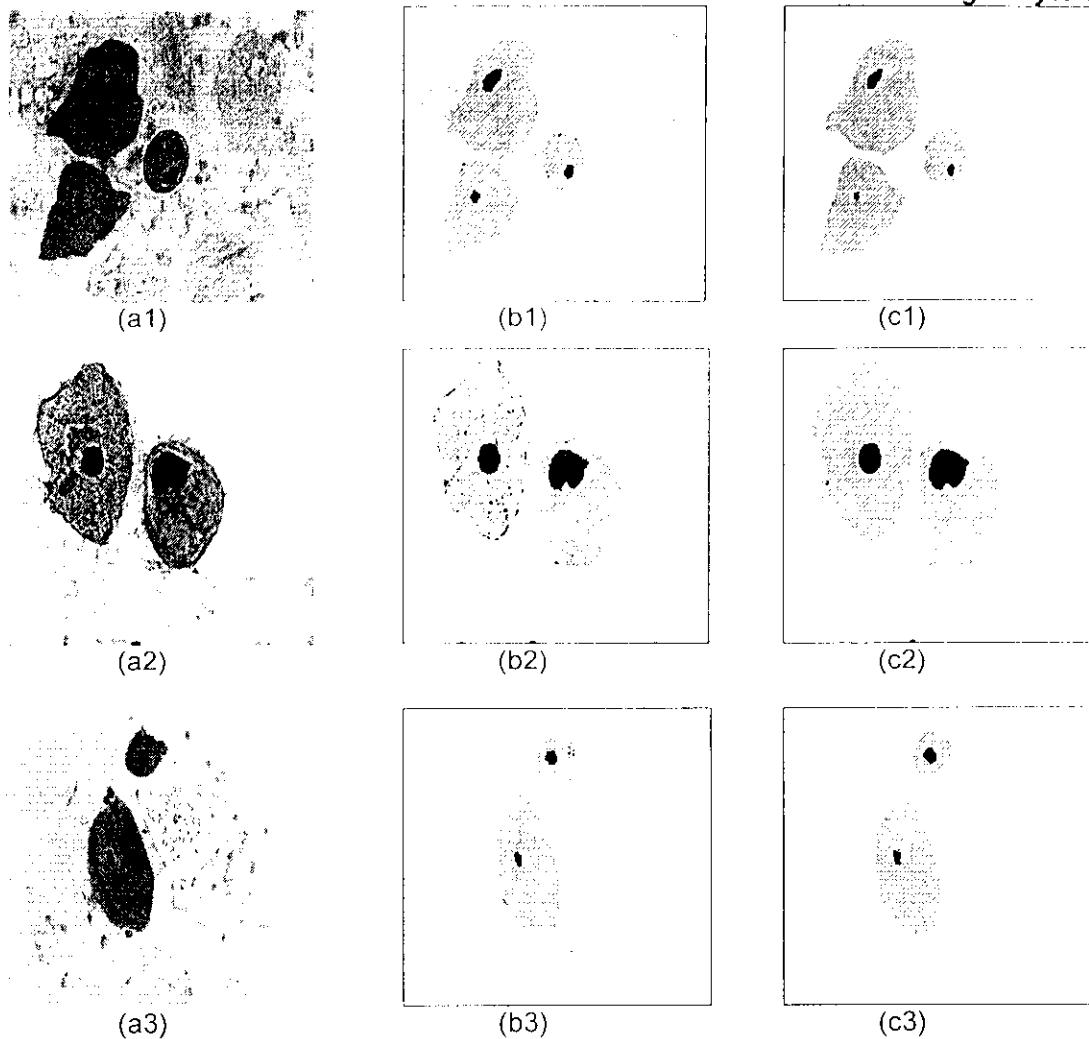


Figure 7 : (ai) Les images originales, (bi) les résultats de segmentation multifractale non supervisée et (ci) les résultats de segmentation multifractale-génétique.

D'après ces résultats, nous constatons une meilleure qualité de segmentation par l'algorithme hybride puisque les pixels sont correctement classés et l'effet de granulosité a diminué. En effet, avec l'algorithme multifractal, chaque point est classé uniquement selon son exposant de singularité qui est estimé localement à son niveau indépendamment de son voisinage. Avec l'étape d'optimisation, chaque point est traité avec son voisinage, et grâce aux opérateurs génétiques, chaque chromosome hérite des meilleures caractéristiques des chromosomes de sa classe. Cependant, la méthode présente un inconvénient majeur, la qualité de segmentation est étroitement liée à la valeur des seuils ($seuil_1$, $seuil_2$) utilisés pour le calcul des fitness ; et nous n'avons aucun critère pour les estimer⁴.

Pour avoir un algorithme de segmentation hybride multifractal-génétique plus souple à l'utilisation, nous proposons de combiner l'algorithme de relaxation sélectionniste avec l'algorithme multifractal. Ce qui permettra de mieux exploiter les résultats de cet AG pour des applications bien spécifiques.

⁴ Les seuils ont des valeurs réelles, non entières

4.4 Algorithme de relaxation multifractal sélectionniste

Cet algorithme découle des caractéristiques suivantes :

Avec la relaxation sélectionniste, même les régions non homogènes d'un même objet sont détectées, cette segmentation malgré juste au sens de détection des zones homogènes, n'est pas exploitable pour la reconnaissance des objets de la scène, puisque le même objet est segmenté en plusieurs zones. Cependant, avec l'algorithme multifractal non supervisé, se sont les grandes classes de l'image qui sont détectées.

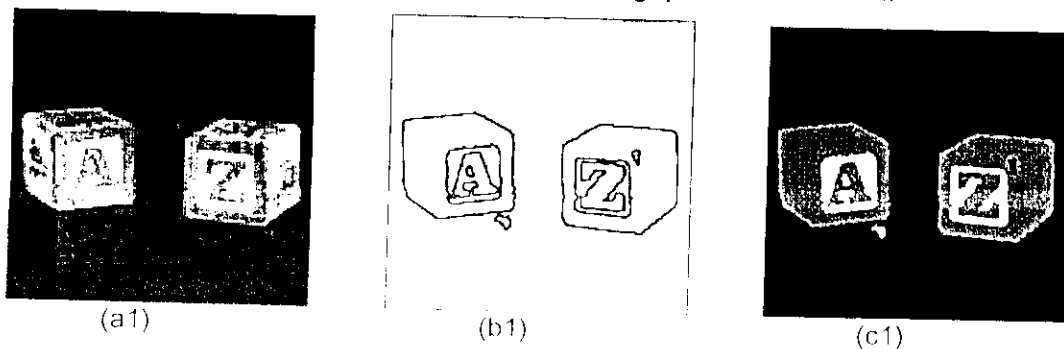
C'est cette propriété de l'algorithme multifractal qui est exploitée pour pallier l'inconvénient de la relaxation sélectionniste. Ainsi, nous proposons l'algorithme de relaxation multifractal sélectionniste, dont les différentes étapes sont :

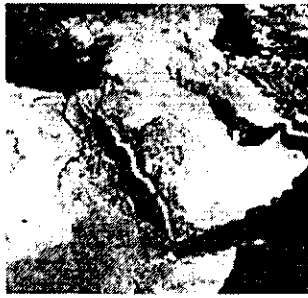
- Calculer les exposants de singularité en chaque pixel de l'image.
- Trouver les intervalles des différentes classes de l'image par l'algorithme non supervisé de segmentation. Dans le cas de la séparation des cellules du col utérin, nous devons estimer deux seuils $S_{\alpha 1}$ et $S_{\alpha 2}$ ($S_{\alpha 1} \neq S_{\alpha 2}$) qui délimitent les trois classes (Noyau, Cytoplasme et Fond).
- Appliquer l'algorithme de relaxation sélectionniste sur l'image originale.
- Estimer la moyenne des niveaux de gris M_{Ri} de chaque région Ri détectée.
- Pour chaque région Ri , calculer sa valeur moyenne de singularité α_{Ri} à partir de M_{Ri} .
- Classer tous les points de la région Ri en comparant sa valeur α_{Ri} aux seuils $S_{\alpha 1}$ et $S_{\alpha 2}$.
- Construire l'image segmentée en affectant une intensité en niveau de gris pour chaque classe $[0, S_{\alpha 1}]$, $[S_{\alpha 1}, S_{\alpha 2}]$ et $[S_{\alpha 2}, 255]$. Et tous les points d'une même région auront l'intensité attribuée à sa classe.

Remarque :

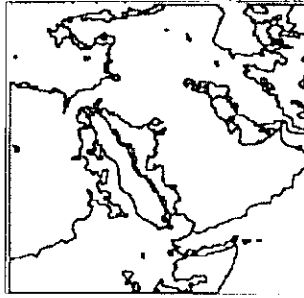
Nous pouvons appliquer l'algorithme de relaxation sélectionniste directement dans l'espace des α de l'image, c.-à-d., qu'il sera appliqué sur les valeurs des singularités des pixels mais non sur leurs valeurs en intensités lumineuses, mais dans ce cas, il faudra d'abord procéder par quantification des valeurs des exposants de singularité.

Dans la figure suivante, nous présentons une série de résultats d'application de l'algorithme de relaxation multifractal sélectionniste. Nous donnons principalement des résultats de segmentation obtenus sur des images de cellules biologiques du col utérin.

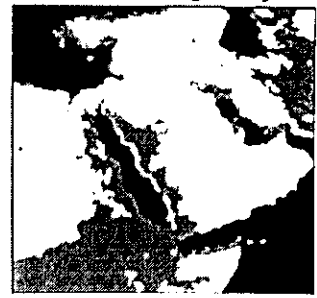




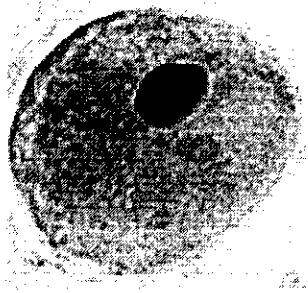
(a2)



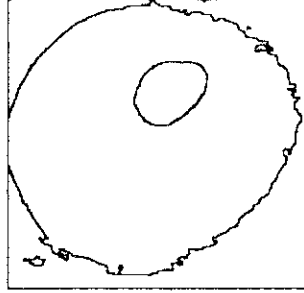
(b2)



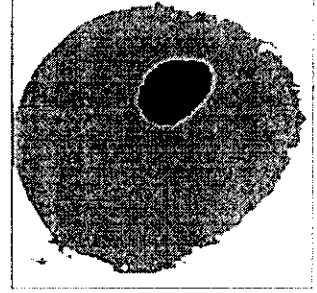
(c2)



(a3)



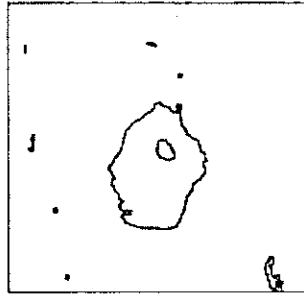
(b3)



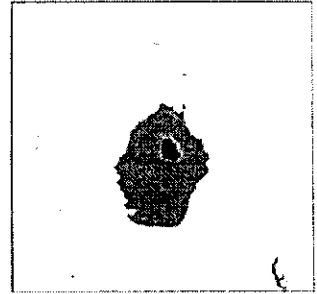
(c3)



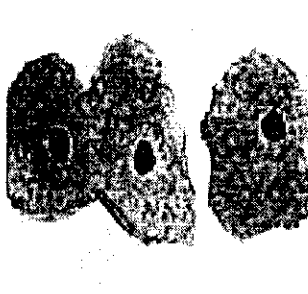
(a4)



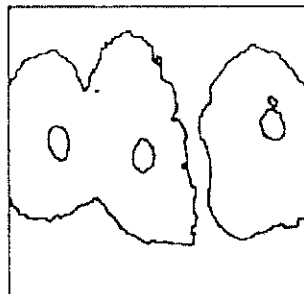
(b4)



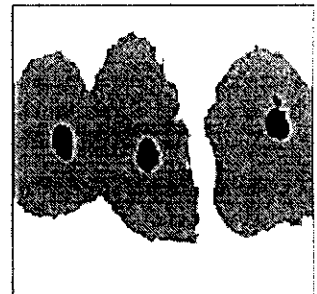
(c4)



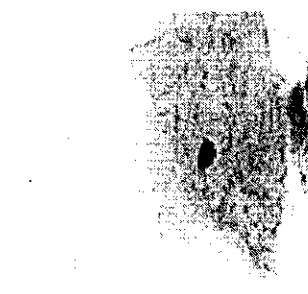
(a5)



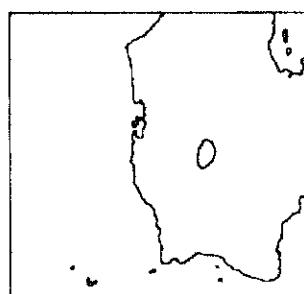
(b5)



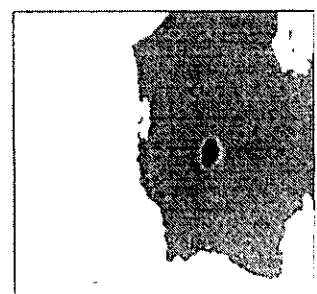
(c5)



(a6)



(b6)



(c6)

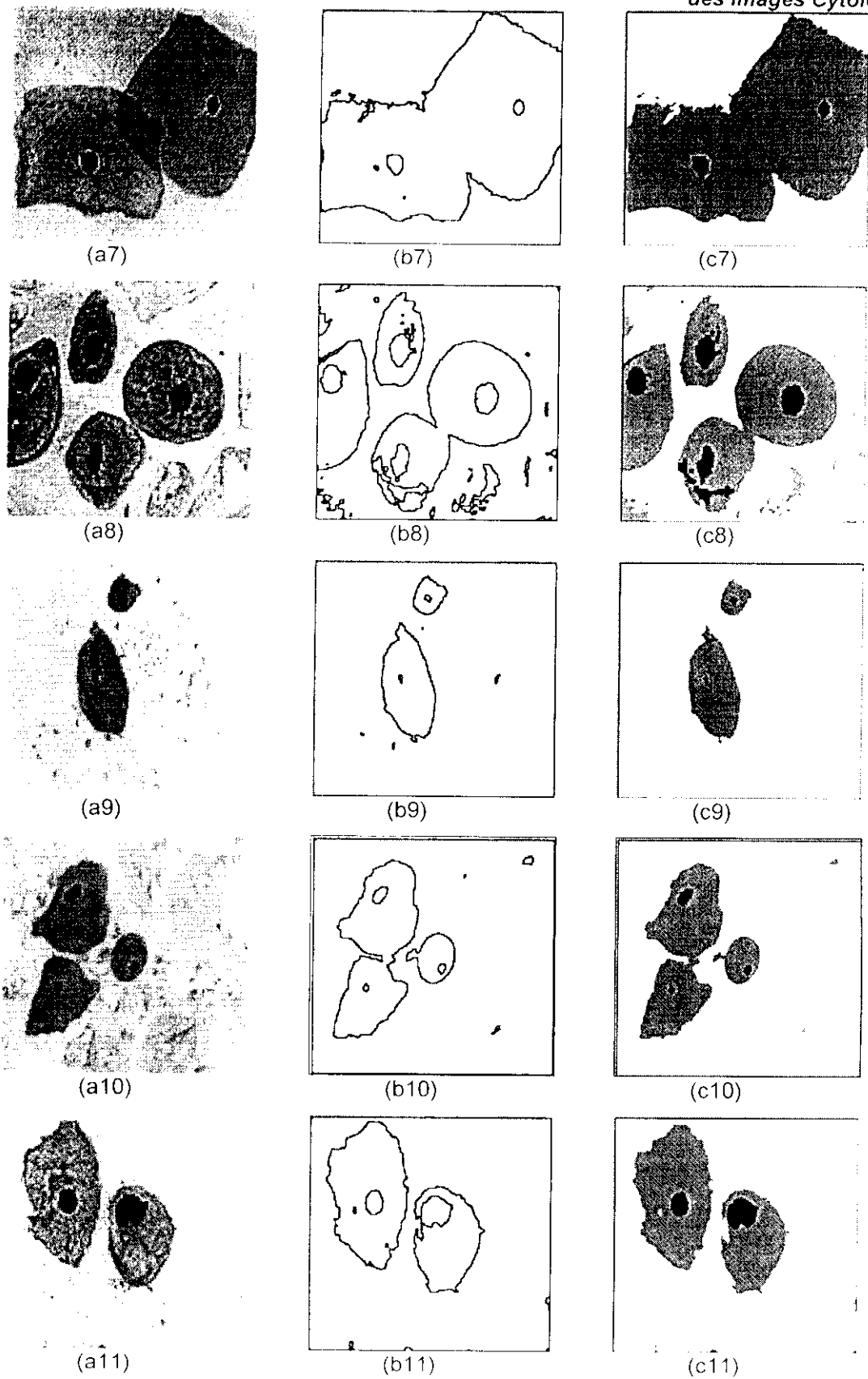


Figure 8 : Segmentation des images par la relaxation multifractale sélectionniste.

Dans le tableau suivant, nous donnons les seuils de singularité $S_{1,1}$ et $S_{1,2}$ utilisés pour obtenir les résultats de la figure 8. Notons que pour certaines images, ces seuils ne

correspondent pas aux seuils estimés par l'algorithme multifractal non supervisé, mais ils sont très proches, l'objectif est d'assurer une meilleure qualité de segmentation.

Numéro de l'image	Seuil S_{α_1}	Seuil S_{α_2}
(a1)	1.821	1.938
(a2)	2.007	2.070
(a3)	2.000	2.123
(a4)	2.026	2.140
(a5)	2.002	2.189
(a6)	1.992	2.209
(a7)	1.978	2.152
(a8)	1.994	2.192
(a9)	2.038	2.215
(a10)	2.026	2.161
(a11)	2.026	2.159

Tableau : Valeurs des seuils S_{α_1} et S_{α_2} des images de la figure 8.

D'après les résultats de la figure 8, nous obtenons une bonne qualité de segmentation par l'algorithme de relaxation multifractal sélectionniste, nous avons réussi à combler les inconvénients de l'algorithme multifractal et adapter l'algorithme de relaxation sélectionniste à la segmentation des objets bien précis de la scène.

Dans l'étude effectuée sur l'algorithme de relaxation sélectionniste, nous avons remarqué que la qualité de la segmentation dépend du nombre de générations N . Pour une faible valeur de N , les limites des régions détectées ne coïncident pas avec les vrais contours des objets de la scène, du fait que les étiquettes qui se propagent de proche en proche n'ont pas pu envahir toute la région à cause de la valeur de N qui n'est pas suffisante. Donc pour un N petit, nous obtenons souvent le même objet de la scène subdivisé en plusieurs régions, donc sur-segmentation. Et pour des grandes valeurs de N , le temps d'exécution devient important.

Avec l'algorithme de relaxation multifractal sélectionniste, la dépendance de la segmentation en fonction de N a été éliminée. En effet, grâce aux intervalles multifractals I_α des différentes classes de la scène, chaque exposant de singularité moyen α_M de la sub-région est estimé et comparé avec les intervalles I_α , et la sub-région est affectée à la classe de l'objet auquel α_M appartient. D'où une reconstitution des objets par comparaison des α_M des sub-régions. Ainsi l'algorithme multifractal remédie au problème du nombre de générations. D'ailleurs, tous les résultats de la figure 8 sont obtenus pour une même valeur de N égale à 50.

Nous concluons que l'algorithme de relaxation multifractal sélectionniste assure une meilleure qualité de segmentation, avec des résultats qui sont plus exploitables pour la reconnaissance puisque les régions segmentées correspondent aux objets de la scène ; et ceci en un temps d'exécution court, ce qui est important puisque les AGs sont généralement consommateurs en temps de calcul.

Dans ce qui suit, nous exploitons les résultats de segmentation obtenus pour effectuer la classification des cellules biologiques du col utérin afin d'organiser la recherche dans l'étape de reconnaissance.

4.5 Classification des cellules biologiques du col utérin [158, 159]

Avec l'algorithme de relaxation multifractal sélectionniste, nous avons pu effectuer la séparation des noyaux et cytoplasmes. Ces corps cellulaires constituent la base de donnée du module de reconnaissance du système de vision par ordinateur pour le dépistage du cancer du col utérin [18, 57].

Et pour mieux organiser le traitement durant l'étape de reconnaissance, nous proposons de classer les cellules du col selon leurs types. Et nous avons choisi le RNC comme critère de classification. Sachant que plus la cellule est jeune et se trouve dans une couche profonde, plus le rapport est élevé.

Nous avons ces intervalles des rapports nucléo-cytoplasmique pour les quatre couches de cellules d'une coupe muqueuse vaginale :

Cellules basales : $0.25 \leq \text{RNC} < 0.37$

Cellules para-basales : $0.15 \leq \text{RNC} < 0.25$

Cellules intermédiaires : $0.04 \leq \text{RNC} < 0.15$.

Cellules superficielles : $0.01 \leq \text{RNC} < 0.04$

Nous effectuons la classification en deux étapes fondamentales qui sont :

- Calcul du rapport nucléo-cytoplasmique de chaque cellule.
- Classification selon le RNC.

4.5.1 Calcul du rapport nucléo-cytoplasmique de chaque cellule.

Nous travaillons sur les images à deux niveaux de gris des noyaux et cytoplasmes obtenues après segmentation par l'algorithme multifractal sélectionniste. Nous appliquons un algorithme d'étiquetage sur chacune des images noyaux et cytoplasmes pour obtenir la surface de chaque corps cellulaire. En effet, l'étiquetage permet de détecter et distinguer chaque région connexe (noyau ou cytoplasme) par affectation d'une étiquette à chacune. Ainsi, la surface de chaque noyau $N_Surface$ ou cytoplasme $C_Surface$ correspond au nombre de points de l'image ayant son étiquette. Et le RNC de chaque cellule est obtenu par le rapport $N_Surface / C_Surface$.

Il est important de noter que pour le calcul du RNC, nous devons effectuer la correspondance entre le noyau et le cytoplasme d'une même cellule. Puisque les étiquettes affectées au noyau et au cytoplasme d'une même cellule peuvent être différentes à cause de leurs positions dans l'image. Sachant que le balayage de l'image se fait de gauche à droite et du haut vers le bas, le noyau peut être la $i^{ème}$ région connexe détectée dans l'image des

noyaux et le cytoplasme peut être la $j^{ème}$ ($i \neq j$) région connexe étiquetée dans l'image des cytoplasmes. Ce qui est une source de conflit et d'erreur pour le calcul du RNC.

Par la figure 9, nous illustrons l'importance de la correspondance des corps cellulaires de chaque cellule. D'après les figures d'étiquetage (d) et (e), le premier noyau et le premier cytoplasme détectés ne correspondent pas à la même cellule (partant du bas vers le haut et de gauche à droite). Sans la correspondance des corps cellulaires, les RNCs estimés seront faux ce qui conduit à une mauvaise classification. D'où l'intérêt de la correspondance des corps cellulaires.

Pour palier ce problème, une correspondance entre les noyaux et les cytoplasmes doit être effectuée. Pour cela, nous devons vérifier que l'emplacement du premier point étiqueté noyau soit voisin aux points du cytoplasme, puisque chaque noyau est entouré de son cytoplasme.

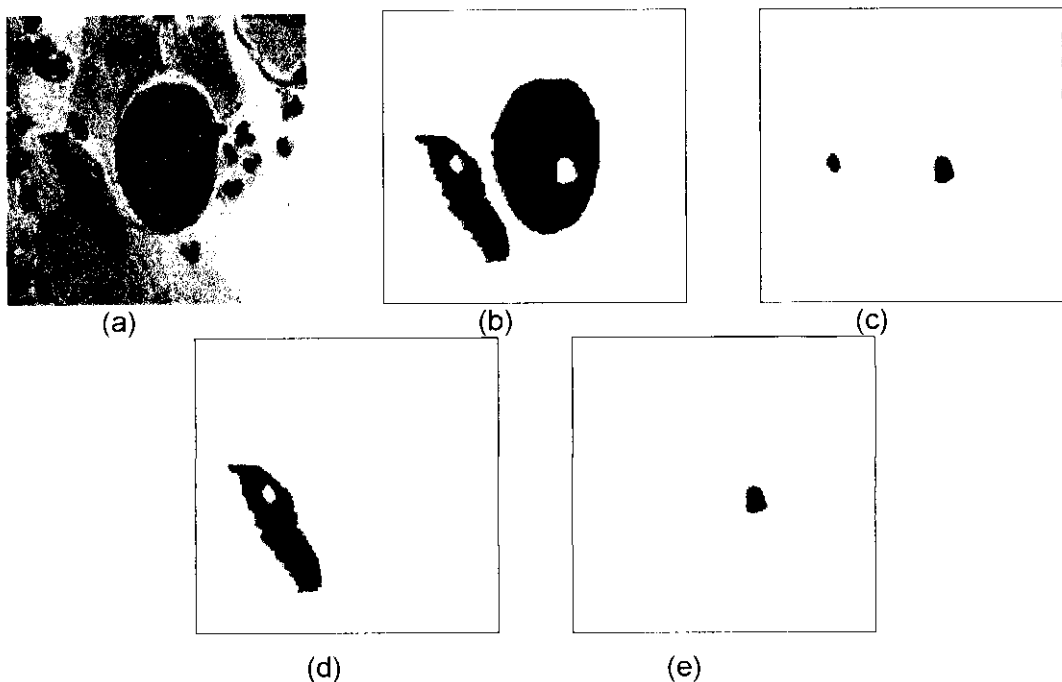


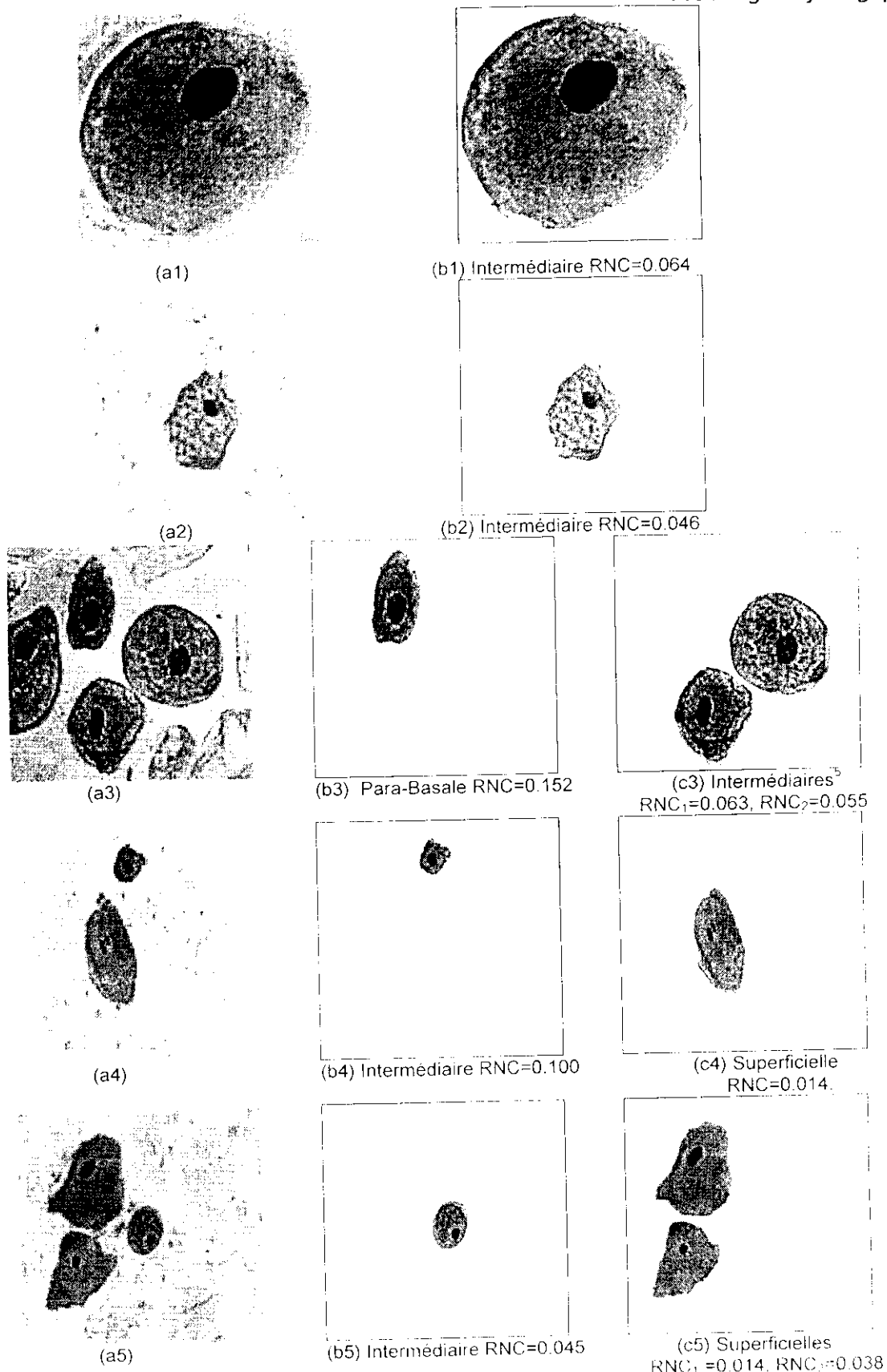
Figure 9: (a) Image originale, (b) et (c) images des cytoplasmes et des noyaux, (d) et (e) cytoplasmes et noyaux étiquetés.

4.5.2 Classification selon le RNC

La classification se fait par comparaison du RNC de chaque cellule avec les intervalles cités précédemment et le résultat est présenté par création de différentes images pour chaque type de cellules.

4.5.3 Application

Par la figure 10, nous donnons une série de résultats de classification des images de cellules biologiques du col. Nous distinguons aisément les noyaux, éléments foncés entourés du cytoplasme, et il est moins facile de distinguer le cytoplasme de chaque cellule. Un noyau et le cytoplasme qui l'entoure forment une cellule et l'image est un amas de cellules.



Le balayage se fait du bas vers le haut et de gauche à droite

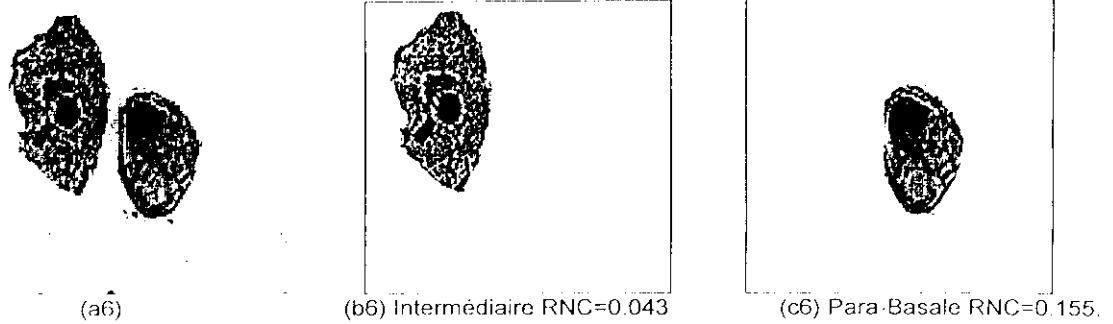


Figure 10 : Classification des cellules du col.

Les résultats obtenus sont satisfaisants, et nous pouvons vérifier la qualité des résultats, sachant que plus la taille du noyau est grande et la taille du cytoplasme petite, la cellule appartient à une couche profonde et vice versa.

Notons que nous pouvons classer une cellule comme étant intermédiaire à cause de son RNC estimé, alors par analyse de sa morphologie, le médecin constate qu'elle est superficielle. Ceci arrive quand la cellule est dans un état inflammatoire qui fait augmenter son RNC.

Pour remédier à cet inconvénient, nous donnons le RNC estimé de chaque cellule et son image. Ainsi, le cytologiste peut vérifier l'état de la cellule et ainsi déduire d'autres informations concernant la présence de la maladie, tel que [21, 160] l'anisonucléose, la difformité nucléaire, l'hyperchromatisme nucléaire et le stade de la maladie. Ce qui est obligatoire dans le dépistage du cancer et donc pour l'étape de reconnaissance.

Remarque : Une meilleure classification se base principalement sur la phase de segmentation. Si des points aléatoires (ou de petites régions) apparaissent, des pré-traitements supervisés doivent être effectués par l'opérateur, et ceci en sélectionnant avec la souris la zone non désirée, et de la filtrer en appliquant une inversion. Mais il faut maintenir précieusement les bords des cellules lors d'élimination des régions inutiles par le prétraitement, puisque la forme des corps cellulaires donne une information sur la présence de la maladie.

Notons aussi que pour des cellules se chevauchant ou tronquées, des pré-traitements doivent être appliqués [18] sinon la classification telle qu'elle est proposée n'est pas valable.

4.6 Conclusion

Dans le cadre de ce chapitre, nous avons utilisé les AGs dans la segmentation des images et exploité les résultats obtenus pour la classification des cellules du col utérin.

Nous avons étudié et appliqué l'algorithme de relaxation sélectionniste. Cette approche affine le résultat du groupement des pixels après chaque génération issue de la population initiale par un ensemble de transformations génétiques de sélection, croisement et mutation.

Pour éviter les variances locales de la mutation, nous avons proposé un autre opérateur qui tient compte des propriétés spatiales de chaque pixel tout en assurant une convergence vers l'optimum global au fur et à mesure des générations en utilisant une fonction de décroissance.

L'une des limitations de la relaxation sélectionniste réside dans l'absence d'un critère d'arrêt fondé sur la stabilité de la segmentation. La méthode telle que proposée consiste à effectuer un nombre de générations suffisamment élevé pour qu'on soit certain que la stabilité ait été atteinte. Cependant, le nombre de générations nécessaires à la convergence dépend en partie du contenu des images à traiter. Il est en outre soumis à la variabilité intrinsèque de la méthode résultant du caractère stochastique de la procédure d'initialisation des unités et des opérateurs génétiques utilisés. Il est donc nécessaire de disposer d'un autre critère que le nombre de générations pour arrêter le programme.

Nous avons alors proposé d'exploiter l'algorithme multifractal non supervisé pour que la segmentation par la relaxation sélectionniste soit plus stable et permette de détecter des objets bien précis de la scène. En effet, par l'algorithme multifractal, les principales classes des objets de la scène sont détectées, ensuite chaque région détectée par l'algorithme de relaxation sélectionniste sera affectée à une des classes multifractales selon son exposant de singularité moyen. Cette solution permet d'assurer une segmentation plus stable de l'image et de détecter les objets de la scène sans pour autant utiliser un N très grand.

De bons résultats sont obtenus pouvant être exploités dans des applications de la vision telle que la reconnaissance de formes.

Nous avons utilisé et exploité la segmentation par étiquetage pour effectuer la classification des cellules biologiques du col utérin. Et ceci en effectuant une analyse morphologique basée sur le calcul du rapport nucléo-cytoplasmique de chaque cellule. La classification est exploitée pour organiser la recherche dans l'étape de reconnaissance du système de dépistage du cancer du col [158, 159].

Conclusion Générale

Conclusion Générale

Cette thèse est née du fait qu'il n'existe pas actuellement de technique universelle de segmentation qui soit en mesure de traiter la multitude d'images réelles qui peuvent apparaître en pratique. Chaque technique particulière est plus ou moins adaptée à un certain contexte. Il se pose donc un problème de choix et de paramétrage des opérateurs en fonction du contexte.

Tout au long de notre travail, nous avons essayé de démontrer la difficulté de la segmentation des images de scènes réelles. Pour espérer une segmentation fiable et suffisamment corrélée avec les entités qui composent la scène réelle, nous avons étudié, implémenté et proposé plusieurs approches qui diffèrent des procédures classiques de segmentation.

Dans cette optique, nous citons nos contributions :

- En plus des techniques classiques, nous avons exploré d'autres théories pour la segmentation, telles que : Fractale, multifractale, morphologie mathématique, la théorie de l'information, la logique floue, ...
- Afin de simplifier l'exploitation de certains algorithmes et assurer une meilleure qualité de segmentation, nous avons proposé des améliorations, tels que les variogrammes pour la détection de contours par les fractales, ou aussi l'utilisation d'un algorithme de seuillage automatique dans la segmentation multifractale.
- Avec les algorithmes génétiques, nous avons une autre manière de segmenter une image sans l'estimation de son gradient ou la recherche des points qui vérifient un critère d'homogénéité. Nous avons étudié l'algorithme de relaxation sélectionniste. Cette approche affine le résultat du groupement des pixels après chaque génération issue de la population initiale par un ensemble de transformations génétiques de sélection, croisement et mutation.
- Pour simplifier l'algorithme de relaxation sélectionniste tout en assurant une bonne qualité de segmentation et surtout une grande adaptabilité pour différents types d'images, nous avons proposé d'utiliser un autre opérateur de mutation qui s'adapte aux propriétés spatiales de chaque pixel.
- Pour assurer la convergence vers un optimum global et éviter que la recherche devienne aléatoire au fur des génération à cause de la mutation, nous avons proposé que la valeur de probabilité de mutation suit une loi de décroissance en fonction de la génération.
- Pour adapter l'algorithme de relaxation sélectionniste à la détection des noyaux et des cytoplasmes des images cytologiques, nous lui avons associé l'analyse

multifractale. Nous avons ainsi proposé deux algorithmes. Le premier procède par optimisation de la segmentation multifractale par classification de pixels par les AGs. De bons résultats sont obtenus mais les paramètres de l'AG sont difficiles à trouver. Et vu, la variabilité du contenu des images, il est encore plus difficile de les estimer. La recherche se fait empiriquement, ce qui est gênant à cause de la consommation exagérée en temps de l'algorithme génétique. La deuxième approche procède par la classification des régions obtenues par la relaxation sélectionniste en utilisant l'algorithme multifractal non supervisé. De meilleurs résultats sont obtenus et l'algorithme est non supervisé.

- Grâce à sa sensibilité aux petites variations d'intensité moyennes des régions et son adaptabilité aux caractéristiques spatiales de l'image, nous avons opté l'algorithme de relaxation multifractal sélectionniste pour la séparation des images de cellules biologiques du col utérin.
- Pour l'organisation de la recherche pour la reconnaissance des anomalies, la segmentation par étiquetage est exploitée pour la classification des cellules biologiques du col utérin selon leurs différents types.
- Tous les algorithmes et traitements étudiés dans le cadre de cette thèse sont implémentés, un logiciel intitulé 'SegPro' d'après 'Segmentation Processing' est conçu sous l'environnement Windows, par le compilateur orienté objet Delphi. 'SegPro' est une interface simple et complète qui permet à l'opérateur une utilisation simple et conviviale quelque soient ses connaissances en informatique.

Les efforts futurs devront porter sur :

- Etude d'autres techniques pour rechercher toujours de meilleures performances.
- Définir une stratégie qui permet, soit d'effectuer la segmentation par adaptation des opérateurs, en sélectionnant de manière adéquate l'opérateur le plus approprié au type d'image, soit par coopération de différentes approches.

Bibliographie

Bibliographie

- [1] R.M. Haralick and L.G. Shapiro, Survey: Image segmentation Techniques, Computer Vision, Graphics and Image Processing, Vol°29, pp: 100-132, 1985.
- [2] K.S. Fu & J.K. Mui, A survey on image segmentation, Pattern Recognition, Vol°13, pp:3-16, 1981.
- [3] Rafael C. Gonzalez, Digital Image Processing, second edition, Addison Wesley, 1987.
- [4] Joan S. Weszka, A survey of Threshold Selection Techniques, Computer Graphics and Image Processing, Vol° 7, pp: 259-265, 1978.
- [5] L. Hamami, N. Lassouaoui, An Approach Fractal and Analysis of Variogram for Edge Detection of biomedical images, 6th International Work Conference on Artificial and Natural Neural Networks IWANN' 2001, Part II, Vol° LNCS 2085, pp: 336-344, June 2001, Spain.
- [6] L. Hamami, N. Lassouaoui, Approche Multifractale appliquée à l'Extraction de Contours et à la Segmentation d'images biomédicales, Algerian Journal of Technologie AJOT, Série B, Vol°14, N°1, pp : 78-85, 1999.
- [7] N. Lassouaoui, L. Hamami, Segmentation des Images de Cellules Biologiques par une Approche Multifractale, 9^{ième} Conférence Internationale sur le Traitement d'Information et Gestion d'Incertitudes dans les systèmes à Base de Connaissance IPMU' 2002, Vol°3, pp: 2005-2011, July 2002, France.
- [8] N. Lassouaoui, L. Hamami, Brain Image Segmentation using mathematical morphology, 2nd IASTED International Conference on Visualization Imaging, and Image Processing VIIP'2002, pp: 318-323, September 2002, Spain.
- [9] N. Lassouaoui, A. Belouchrani, L. Hamami, On the Use of Multi-fractal Analysis and Genetic Algorithms for the Segmentation of Cervical Cell Images, International Journal of Pattern Recognition and Artificial Intelligence, World Scientific Publishing Company, Vol°17, N°7, pp: 1227-1244, 2003.
- [10] N. Lassouaoui, L. Hamami, F. Chehbour, Supervised Segmentation of the Cervical Cell Images by using the Genetic Algorithms, International Work-conference on Artificial and Natural Neural Networks IWANN'2003, Vol° LNCS 2687, pp: 385-392, June 2003, Espagne.
- [11] D. E. Goldberg, Algorithmes Génétiques : Exploration, optimisation et apprentissage automatique, Edition Wesley, 1989.
- [12] John. H. Holland, Genetic Algorithms, pour la science, Edition of Scientific American, N°179, pp:44-50, July 1992.
- [13] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms: Concepts and Applications, IEEE Transactions on Industrial Electronics, Vol°43, N°5, pp: 519-534, 1996.
- [14] L.M. Schmitt, Fundamental study: Theory of genetic algorithms, Theoretical Computer Science, Vol°259, pp: 1-61, 2001.
- [15] P. Andrey, Selectionist relaxation: genetic algorithm applied to image segmentation, Image and vision computing, Vol°17, pp: 175-187, 1999.
- [16] P. Andrey & P. Tarroux, Unsupervised Segmentation Using a Distributed Genetic Algorithm, Pattern Recognition, Vol°27, N°5, pp: 659-673, 1994
- [17] P. Andrey, Segmentation d'images par algorithmes génétiques Thèse de doctorat Sciences appliquées. Paris 7, 1997.

- [18] N. Lassouaoui, L. Hamami, A. Zerguerras, Segmentation and classification of biological cell images by a multifractal approach, *International Journal of Intelligent systems*, Vol°18, N°6, pp: 657-678, Wiley Edition, June 2003.
- [19] C. Garbay, Modélisation de la couleur dans le cadre de l'analyse d'images et de son application à la cytologie automatique, Thèse de doctorat, Institut National Polytechnique de Grenoble, 1979.
- [20] Alain Boucher, Une approche décentralisée et adaptative de la gestion d'informations en vision, Application à l'interprétation d'images de cellules en mouvement, Thèse de Doctorat, Université Joseph Fourier, Grenoble1, Janvier 1999.
- [21] C. Gompel, Atlas de la cytologie, Ecole de Médecine de Paris, 1982.
- [22] J.P. Cocquerez et S. Philipp, Analyse d'images : filtrage et segmentation, Edition Masson, 1995.
- [23] J.M. Salotti, Gestion des informations dans les premières étapes de la vision par ordinateur, Thèse de doctorat de l'Institut National Polytechnique de Grenoble, Janvier 1994.
- [24] G. Braviano, Logique floue en segmentation d'images : Seuillage par entropie et structures pyramidales, Doctorat de l'Université Joseph Fourier, Grenoble I, 1995.
- [25] Y.J. Zhang, A survey on evaluation methods for image segmentation, *Computer Vision and Pattern Recognition*, Elsevier Science, Vol°29, N°8, pp: 1335-1346, 1996.
- [26] Y.J. Zhang, A review of recent evaluation methods for image segmentation, *The sixth International Symposium on IEEE Signal Processing and its Applications*, pp: 148-151, Août 2001, Malaysia.
- [27] C. Spinu, Une approche multi-agents pour la segmentation d'images associant estimation et évaluation, Thèse de Doctorat, Université Joseph Fourier, Grenoble 1, 1997.
- [28] M.D. Levine, A.M. Nazif, Dynamic measurement of computer generated image segmentations, *IEEE transactions on Pattern Analysis and Machine Intelligence*, Vol°7, N°2, pp: 155-164, 1985.
- [29] F.L. Valverde, N. Guil, J. Munoz, R. Nishikawa, K. Doi, An evaluation criterion for edge detection techniques in noisy images, *International Conference on Image Processing*, vol°1, pp: 766-769, 2001, Greece.
- [30] T. Pavlidis, Why progress in machine vision is so slow, *Pattern recognition letters*, Vol°13, pp : 211-225, 1992.
- [31] D. Harwood, M. Subbarao, H. Hakalahti, L.S. Davis, A new class of edge-preserving smoothing filters, *Pattern recognition Letters*, Vol°6, pp: 155-162, 1987.
- [32] J. Canny, A computationnal approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol°8, N°6, pp :679-698, Nov 1986.
- [33] C. Garbay, Image Structure Representation and Processing : A discussion of Some Segmentation Methods in Cytology, *IEEE Transactions on Pattern and Machine Intelligence*, Vol° 8, N° 2, pp: 140-146, Mars 1986.
- [34] R. Ohlander, K. Price, D.R. Reddy, Picture segmentation using a recursive splitting method, *Computer Graphics and Image Processing*, Vol°8, pp: 313-333, 1978.
- [35] S. Shafer, T. Kanade, Recursive region segmentation by analysis of histogram, *International IEEE Conference on Acoustics, Speech and Signal Processing*, pp: 1166-1171, 1982.
- [36] Fabrice Bellet, Une approche incrémentale à base de processus coopératifs et adaptatifs pour la segmentation des images en niveaux de gris, Thèse de Doctorat, Institut National Polytechnique de Grenoble. 1998.
- [37] J. Kittler & J. Illingworth, Relaxation labelling algorithms – a review, *Proceeding on Image and Vision Computer*, vol°3, N°4, pp: 206-216, November 1985.

- [38] A. Rosenfeld, R.A. Hummel, S.W. Zucker, Scene Labeling by Relaxation Operations, IEEE Transactions on Systems, Man and Cybernetics, Vol° 6, N°6, pp: 420-433, Juin 1976.
- [39] J-P. Wang, Stochastic relaxation on partitions with connected components and its applications to image segmentation, IEEE transactions on Pattern Analysis and Machine Intelligence, Vol°20, N°6, pp: 619-636, Jun 1998.
- [40] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian Restoration of images, IEEE transactions on Pattern Analysis and Machine Intelligence, Vol°6, N°6, pp: 721-741, Nov 1984.
- [41] R. M. Haralick, Statistical and structural approaches to texture, Proceeding of the IEEE, Vol°67, N°5, pp: 786-804, May 1979.
- [42] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE transactions on systems, man and cybernetics, Vol°3, N°6, pp: 610-621, Nov 1973.
- [43] C. D. Kermad, Segmentation d'images : Recherche d'une mise en œuvre automatique par coopération de méthodes, thèse de doctorat de l'Université de Rennes I, 1997.
- [44] R. M. Haralick, Image analysis using mathematical morphology, Transactions on Pattern and Machine Intelligence, Vol°9, pp: 532-550, 1987.
- [45] Afcet, 8^{ième} Congrès, Reconnaissance des formes et intelligence artificielle, Vol° 2, pp : 848-857, 25-29 Nov 1991, Lyon, Villeurbanne.
- [46] Serge Beucher, Segmentation d'images et Morphologie Mathématique, Thèse de Doctorat, Ecole Nationale des Mines de Paris, 1990.
- [47] J.M. Keller, S. Chen, R. Crownover, Texture description and segmentation through fractal geometry, Computer Graphics and Image Processing, Vol°45, pp: 150-166, 1989.
- [48] H. E. Schepers, J. H. G. M. van Beek et J. B. Bassingthwaighte, Four Methods to Estimate The fractal Dimension from Self-Affine Signals, IEEE Engineering in Medicine And Biology, Vol°11, N°2, pp : 57-64, juin 1992.
- [49] L. Hamami, N.Lassouaoui, L. Mahiddine, Différentes Méthodes d'Estimation des paramètres fractals : Application aux images, Conférence méditerranéenne Fractales'2000, pp : 39-44, 11-12 Nov. 2000, Constantine.
- [50] Y. Liu, Y. Li, Image Feature Extraction and Segmentation using Fractal Dimension, IEEE Conference on Information Communications and Signal Processing, pp :975-979, septembre 1997, Singapour.
- [51] J.P. Berroir, Application des mesures multifractales à l'analyse et la segmentation d'images médicales, Rapport d'INRIA, N°1345, pp : 979-987.
- [52] J. Levy-Vehel, J.P. Berroir, Image analysis through multifractal description, Rapport de Recherche INRIA, projet Epidaure, N°1942, pp : 1-17, Juin 1993.
- [53] B. Wrobel, O. Monga, Segmentation d'images naturelles: Coopération entre un détecteur de contour et un détecteur de region, Actes du 11^{ème} colloque GRETSI, N°9, pp: 21-30, 1977.
- [54] Catherine Garbay, Images, Stratégies Perspectives et Stratégies Cognitives d'Analyse, Thèse de Doctorat, Institut National Polytechnique de Grenoble, 1986.
- [55] B. Mandelbrot, Les objets fractals, Flammarion, 1995.
- [56] B. Mandelbrot, Fractals Form, Chance, And Dimension, W.H. Freeman And Company, 1977.
- [57] N. Lassouaoui. Segmentation des images biomédicales par des approches fractales. Conception d'un logiciel de traitement et de segmentation des images. Thèse de Magister, Ecole Nationale Polytechnique d'Alger, 2000.
- [58] J.M. Blackledge and E. Fowler, Fractal Dimension Segmentation of Synthetic Aperture Radar Images. International Conference on Image Processing and its Applications, pp 445 449, 1992.

- [59] C.J. Burdett et M. Desai., Localised fractal dimension measurement in digital mammographic images, SPIE, Vol° 2094, pp: 141-151, 1993.
- [60] L. Mahiddines, Contribution à l'utilisation des fractales en stéréovision, Thèse de Magister, Institut d'électronique, USTHB, 1998.
- [61] A. Arneodo, F.Argoul, E.Bacry, J. Elezgaray, J-F. Muzy, Ondelettes, multifractales et turbulences de l'ADN aux croissances cristallines, Arts et Sciences, 1995.
- [62] S. Soltane, J. Claude Angue, Sélection d'opérateurs directionnels basée sur les variogrammes, Revue internationale es technologies avancées, N°11, pp: 14-22, Juillet 1999.
- [63] Michel Schmitt, Des Algorithmes Morphologiques à l'Intelligence Artificielle, Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Paris, 1989.
- [64] F. Meyer et S. Beucher, Morphological Segmentation, Journal of Visual Communication and Image Representation, Vol° 1, N° 1, pp : 21-46, Sept 1990.
- [65] Christophe Laurent, Conception d'algorithmes parallèles pour le traitement d'images utilisant la morphologie mathématique. Application à la segmentation d'images, Thèse de Doctorat, Ecole Doctorale de Mathématiques et Informatique, 1998.
- [66] A. Meijster & J. B. T. M. Roerdink, Computation of watersheds based on parallel graph algorithms, Mathematical Morphology and its applications to image and signal processing, ISMM'96, pp: 305-312, 1996, Atlanta.
- [67] Serge Beucher, Watershed Hierarchical Segmentation and Waterfall Algorithm, Mathematical Morphology and its Application to Image Processing, Kluwer Academic Publishers, pp: 69-76, 1994.
- [68] S. Bres, P.A. Champin, J.M. Heraud, V. Herilier, J.M. Jolion, E. Loupias, Traitement d'images et vision par ordinateur, Projet TELESUN, A world wide multimedia TELEteaching System for Universities, <http://telesun.insa-lyon.fr/~telesun/welcome.html>.
- [69] S. Beucher & Meyer, The Morphological Approach to Segmentation: The Watershed Transformation, pp: 433-481, Chap 12, extrait de Mathematical Morphology in Image Processing, Edward R. Dougherty edition, 1992.
- [70] M. Lezoray Olivier, Segmentation d'images couleur par morphologie mathématique et classification de données par réseaux de neurones : application à la classification de cellules en cytologie des séreuses, Thèse de Doctorat, Université de CAEN/ Basse Normandie, 2000.
- [71] N.E. Davison, H. Eviatar, R.L. Somorjai, Snakes simplified, Pattern recognition, Vol°33, pp: 1651-1664, 2000.
- [72] T.F. Chan, L.A. Vese, Active contours without edges, IEEE transaction on image processing, Vol°10, N°2, pp: 266-277, Février 2001.
- [73] Y.Y. Wong, P.C. Yuen, C.S. Tong, Segmented snake for contour detection, Pattern recognition, Vol°31, N°11, pp: 1669-1679, 1998.
- [74] H. L. Tan, S. B. Gelfand, E. J. Delp, A cost minimization approach to edge detection using simulated annealing, Transactions on Pattern and Machine Intelligence, Vol°14, pp: 3-18, janvier 1991.
- [75] Z. Dokur, T. Ölmez, Segmentation of ultrasound images by using a hybrid neural network, Pattern recognition letters, Vol°23, pp: 1825-1836, 2002.
- [76] T. Pun, Entropic thresholding, A new approach, Computer graphics and image processing, Vol°16, pp: 210-239, 1981.
- [77] J. N. Kapur, P.K. Sahoo, A.K.C. Wong, A new method for gray-level picture thresholding usinf the entropy of histogram, Computer Vision Graphics and Image Processing, Vol°29, pp: 273-285, 1985.

- [78] K.V. Mardia, T.J. Hainsworth, A spatial thresholding method for image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol°10, N°6, pp: 919-927, Nov 1996.
- [79] S. Guillaume, Introduction de règles floues interprétables, Doctorat de l'institut National des Sciences Appliquées, Toulouse, 2001.
- [80] F. Wong, R. Nagarajan, S. Yaacob, A. Chekima, N. Belkhamza, An image segmentation method using fuzzy-based threshold, *The sixth International Symposium on IEEE Signal Processing and its Applications*, pp: 144-147, Août 2001, Malaysia.
- [81] T. D. Pham, Image segmentation using probabilistic fuzzy C-means clustering, *International Conference on Image Processing*, vol°1, pp:722-725, 7-10 Oct. 2001 Thessaloniki, Greece.
- [82] W. Pedrycz, Fuzzy sets in pattern recognition: methodology and methods, *Pattern recognition*, Vol°23, pp: 121-146, 1990.
- [83] J.M. Chassery & C. Garbay, An Iterative Segmentation Method Based on a Contextual Color and Shape Criterion, pp: 794-800, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol°6, N°6, pp: 794-800, Nov 1984.
- [84] J. M. Chassery & C. Garbay, Iterative process for color image segmentation using a convexity criterion, *SPIE* 397, pp: 165-172, 1983.
- [85] P.W. Fung, K.K. Ly, Y. Attikiouzel, Automatic Segmentation of Biomedical Images, *IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Vol°2, pp: 882-885, 1988.
- [86] S. Peleg, A new Probabilistic Relaxation Scheme, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol° 2, N° 4, pp: 362-369, Juillet 1980.
- [87] N. Lassouaoui, L. Hamami, La relaxation probabiliste non supervisée dans la segmentation des images de cellules biologiques, *International Conference: Sciences of Electronic, Technologies of Information and Telecommunications SETIT'2004*, 15-20 Mars 2004, Tunisie.
- [88] P. Scheunders, A genetic Lloyd-Max image quantization algorithm, *Pattern Recognition Letters*, Vol°17, N°5, pp: 547-556, May 1996.
- [89] P. Scheunders, A genetic approach towards optimal color image quantization, *IEEE International Conference on Image Processing*, Vol°3, pp: 1031-1033, Sept 1996, Switzerland.
- [90] H. Peng, F. Long, Z. Chen, D. D. Feng, W. Siu, Hierarchical genetic image segmentation algorithm based on histogram dichotomy, *Electronics-Letters*, Vol°36, N°10, pp: 872-874, May 2000.
- [91] H. Peng, F. Long, Z. Chi, W. Su, A hierarchical distributed genetic algorithm for image segmentation, *Proceedings of Congress on Evolutionary Computation CEC'2000*, Vol°1, pp: 272-276, 2000, USA.
- [92] H. J. Kim, E. Y. Kim, J. W. Kim, S. H. Park, MRF model based image segmentation using hierarchical distributed genetic algorithm, *Electronic Letters*, Vol°34, N°25, pp: 2394-2395, Dec 1998.
- [93] E. Y. Kim, J. W. Kim, S. H. Park, H. J. Kim, M. H. Park, An MRF model based image segmentation using hierarchical distributed genetic algorithm, *Second IEEE International Conference on Intelligent Processing Systems*, pp: 438-442, 1998, Australia.
- [94] Y. Shik Moon & T. Hycon Kim, Efficient morphological segmentation using a new connector operator, *Electronics Letters*, Vol° 36, N° 1, pp : 22-24, January 2000.
- [95] J.P. Thiran, B. Macq & J. Mairesse, Morphological Classification of Cancerous Cells, *IEEE International Conference on Image Processing*, Vol° 3, pp : 706-710, 1994.
- [96] M. Pesaresi, J.A., Benediktsson, A new approach for the morphological segmentation of high-resolution satellite imagery, *IEEE Transactions on Geoscience and Remote Sensing*, Vol° 39, N°2, pp: 309-320, Feb 2001.

- [97] N. Lassouaoui, L. Hamami, N. Nouali, B. Ait Abdelkader, Extraction des lésions dans des images de scanner cérébral pour système de télé-diagnostic, Edité par Groupe des Ecoles des Télécommunications et Hermes-Lavoisier, Vol°59, Janvier-Février 2004.
- [98] J.P. Postaire, De l'image à la décision, édition Dunod, 1987.
- [99] L. Mahiddine, M. Benkhelif, O. Djekoune, Un algorithme d'étiquetage de régions dans une image, Séminaire sur l'imagerie et la transmission SIT'99, pp : 13-21, 30 Nov-1Déc 1999, Algérie.
- [100] I. Abdel Salam, W. El Haweet and E. Matter, A survey on Genetic Algorithms Concepts and Operators, Computer & Automatic Control Dept. University of Alexandria, pp:67-72.
- [101] A. Petrowski, Alain Pétrowski, Une introduction à l'optimisation par algorithmes génétiques, <http://www-inf.int-evry.fr/~ap/EC-tutoriel/Tutoriel.html>.
- [102] K. De Jong, Learning with Genetic Algorithms: An Overview, Machine Learning, Vol° 3, pp: 121-138, 1988.
- [103] G. Syswerda, Uniform crossover in Genetic Algorithms, 3rd International Conference on Genetic Algorithms, pp: 2-9, 1989, USA.
- [104] I.C. Lerman & R. F. Ngouenet, Algorithmes génétiques séquentiels et parallèles pour une représentation affine des proximités, Rapport de Recherche d'INRIA, N° 2570, Janvier 1995.
- [105] S. M. Bhandarkar, Y. Zhang, W. D. Potter, An edge detection technique using genetic algorithm based optimisation, Pattern Recognition, Vol°27, N°9, pp: 1159-1180, 1994.
- [106] S. M. Bhandarkar, H. Zhang, Image segmentation using evolutionary computation, IEEE Transactions on Evolutionary Computation, Vol°3, N°1, pp: 1-21, . April 1999.
- [107] A. Calle, W. D.Potter, S. M. Bhandarkar, A genetic algorithm for image segmentation, Ninth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp: 745-750, 1996, Japan
- [108] H. L. Tan, S. B. Gelfand, E. J. Delp, A comparative cost function approach to edge detection, IEEE Transactions on Systems, Man, and Cybernetics, Vol°19, N°6, pp:1337-1349, 1989.
- [109] B. Bhanu, S. Lee, J. Ming, Self-Optimizing image segmentation system using a genetic algorithm, Proceeding of the 4th International Congress on Genetic Algorithms, pp: 362-369, 1991.
- [110] B. Bhanu, S. Lee and J. Ming, Adaptive Image segmentation Using a Genetic Algorithm, IEEE Transactions on Systems, Man, And Cybernetics, vol°25, N° 12, 1995.
- [111] B. Bhanu, S. Lee, S. Das, Adaptive image segmentation using genetic and hybrid search methods, IEEE transactions on Aerospace and Electronic Systems, Vol°31, N°4, pp:1268-1291, Oct 1995.
- [112] B. Bhanu, S. Fonder, Learning based interactive image segmentation, 15th International Conference on PatternRecognition, vol°1, pp: 299-302, Sept. 2000, Spain.
- [113] Z. Dokur, T. Olmez, Segmentation of MR and CT images by using a quantiser neural network, Journal of Neural Computing Applications, Springer-Verlag, Vol°11, pp: 168-177, 2003.
- [114] K. Matsui, Y. Kosugi, New image segmentation method by modified counter-propagation network and genetic algorithm, IEEE International Conference on Systems, Man, and Cybernetics, vol°4, pp: 854-858, 12-15 Oct 1999 Tokyo, Japan
- [115] K. Matsui, Y. Kosugi, Image segmentation by neural-net classifiers with genetic selection of feature indices, International Conference on Image Processing, vol°1, pp. 524-528, Oct 1999, Japan
- [116] L. Ballerini. Genetic snakes for color images segmentation. Workshops on Applications of Evolutionary Computing, Vol° LNS 2037, pp. 268-77, 2001

- [117] Y. Xu, S.C. Neu, C.J. Ornes, J. F. Owens, J. Sklansky, D.J. Valentino, Optimization of active-contour model parameters using genetic algorithms: segmentation of breast lesions in mammograms, Proceedings of the SPIE International Society for Optical Engineering, pp: 1406-1414, 21-28 Feb. 2002, San Diego.
- [118] T. Jiang, Y. Fan, Parallel genetic algorithm for 3D medical image analysis, IEEE International Conference on Systems, Man and Cybernetics, Vol°6, 2002, Tunisia.
- [119] Y. Fan, T. Jiang, D. J. Evans, Volumetric segmentation of brain images using parallel genetic algorithms, IEEE-Transactions-on-Medical-Imaging, Vol°21, N°8, pp: 904-909, Aug. 2002.
- [120] T. Jiang, F. Yang, An evolutionary tabu search for cell image segmentation, IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol°32, N°5, pp: 675-678, Oct 2002.
- [121] A. M. Alimi, Evolutionary computation for the recognition of on-line cursive handwriting, IETE Journal of Research, vol°78, N°5, pp: 385-396, 2002.
- [122] S. Wada, N. Imai, Efficient structural segmentation of scene image with genetic algorithm for merging boundary regions, IASTED International Conference on Signal and Image Processing, pp: 33-37, Aug 2001, Honolulu.
- [123] G. Yen, N. Nithianandan, Facial feature extraction using genetic algorithm, Congress on Evolutionary Computation, vol°2, pp: 1895-1900, 2002.
- [124] K. Takagi, A. Koike, S. Matsumoto, H. Yamamoto, Motion picture coding based on region segmentation using genetic algorithm, Systems and Computers, Vol°33, N°5, pp: 41-50, 2002, Japan.
- [125] I.V. Maslov, I. Gertner, Efficient image reduction for image registration with evolutionary algorithm, SPIE International Society for Optical Engineering, Vol°4787, pp: 198-209, 2002, Seattle, WA, USA.
- [126] K. Jung, J. Han, Stereo correspondence using GA-based segmentation, Vol° LNCS 1983, pp: 497-502, 2000.
- [127] V. Ramos, F. Muge, Map segmentation by colour cube genetic k-Mean clustering, Fourth European Conference on Research and Advanced Technology for Digital Libraries, Vol° LNCS 1923, pp: 319-230, 18-20 Sept. 2000, Lisbon, Portugal.
- [128] M. Abdulghafour, A. Fellah, A hybrid approach of genetic algorithms and fuzzy logic applied to feature extraction from multisensory images, IEEE International Conference on Systems, Man, and Cybernetics, Vol°1, pp: 199-204, 1997, USA.
- [129] H. Zheng, L. X. Kong, S. Nahavandi, Automatic inspection of metallic surface defects using genetic algorithms, 9th International Conference on Metal Forming, Vol° 125-126, pp: 427-333, Sept 2002, Birmingham, UK.
- [130] E. Lutton, P. Martinez, Détection de primitives géométriques bidimensionnelles dans les images à l'aide d'un algorithme génétique, Rapport de recherche Robotique, image et vision, INRIA, N° 2110, Nov 1993.
- [131] E. Lutton, P. Martinez, A genetic algorithm for the detection of 2D Geometric Primitives in images, Computer Vision and Image Processing, Vol°1, pp: 526-528, 1994.
- [132] S. Kahlouche, K. Achour, O. Djekoune, A genetic algorithm for geometric primitives extraction, IEEE Seventh International Symposium on Signal Processing and its Applications Vol° 1, pp: 509-512, July 1-4 2003, Paris, France.
- [133] D. N. Chun, H. S. Yang, Robust image segmentation using genetic algorithm with a fuzzy measure, Pattern-Recognition, Vol°29, N°7, pp: 1195-1211, 1996.
- [134] C. Rosenberger, Mise en oeuvre d'un système adaptatif de Segmentation d'images, Thèse de doctorat, Université de Rennes, 1999.
- [135] C. Rosenberger, K. Chehdi, Supervised genetic image segmentation. IEEE International Conference on Systems, Man and Cybernetics, vol°5, 6-9 Oct 2002, Tunisia.

- [136] C. Bounsaythip, J. Alander, Genetic algorithms in image processing-a review, Third Nordic Workshop on Genetic Algorithms and their Applications, pp: 173-192, 20-22 Aug. 1997, Finland.
- [137] P. Zingaretti, A. Carbonaro, P. Puliti, Evolutionary image segmentation, 9th International Conference on Image Analysis and Processing, vol^o1, pp: 247-254, 17-19 Sept. 1997, Italy.
- [138] A. Visa, A genetic algorithm based method to improve image segmentation, Fourteenth International Conference on Pattern Recognition, vol^o2, pp: 1015-1017, 16-20 Aug. 1998, Australia.
- [139] M. Haseyama, M. Kumagai, H. Kitajima, A genetic algorithm based image segmentation for image analysis, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP99, vol^o6, pp: 3445-3448, March 1999, USA
- [140] W. Lei, S. Tingzhi, Improved adaptive genetic algorithm and its application to image segmentation, The International Society for Optical Engineering, vol^o4550, pp: 115-120, 2001.
- [141] W. Lei, S. T. Zhi, Two-dimensional entropy method based on genetic algorithm, Journal-of Beijing Institute of Technology, vol^o11, N^o2, pp: 184-188, June 2002.
- [142] G. L. Bosco, A genetic algorithm for image segmentation, 11th International Conference on Image Analysis and Processing, pp: 262-266, 26-28 Sept. 2001, Palermo, Italy,.
- [143] S. Y. Ho, K. Z. Lee, An efficient evolutionary image segmentation algorithm, Congress on Evolutionary Computation, vol^o2, pp: 1327-1334, 27-30 May 2001, South Korea.
- [144] H. Y. Shinn, K. Z. Lee, Design and analysis of an efficient evolutionary image segmentation algorithm, Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, vol^o35, N^o1, pp: 29-42, Aug. 2003.
- [145] M. Yoshimura, S. Oe, Texture image segmentation by genetic algorithms, IEEE International Conference on Evolutionary Computation, pp: 125-130, 20-22 May 1996, Japan.
- [146] M. Yoshimura, S. Oe, Edge detection of texture image using genetic algorithms, 36th International IEEE Annual Conference, pp: 1261-1266, July 1997, Japan.
- [147] M. Yoshimura, S. Oe, Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas, Pattern Recognition, Vol^o32, N^o12, pp: 2041-2054, Dec. 1999.
- [148] M. Yoshimura, S. Oe, Evolutionary segmentation and multiresolution analysis of texture image using genetic algorithms and two-dimensional wavelet decomposition, 5th International Conference on Soft Computing and Information/Intelligent Systems, Vol^o2, pp: 987-991, 1998.
- [149] J.S. Pan, J.W. Wang, Texture segmentation using separable and non-separable wavelet frames, Transactions on Fundamentals of Electronics, Communications and Computer Sciences, pp: 1463-1474, Aug. 1999.
- [150] B. K. Jeon, J. H. Jang, K. S. Hong, Road detection in SAR images using genetic algorithm with region growing concept, International Conference on Image Processing, Vol^o2, pp: 688-691, Sept 2000, Canada.
- [151] M. Yu, N. Eua-anant, A. Saudagar, L. Udpa, Genetic algorithm approach to image segmentation using morphological operations, International Conference on Image Processing, vol^o3, pp: 775-779, 1998, USA
- [152] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern recognition, Vol^o33, pp: 1455-1465, 2000.
- [153] D. L. Swets, B. Punch, J. Weng, Genetic algorithms for object recognition in a complex scene, International Conference on Image Processing, Vol^o2, pp: 595-598, 1995.
- [154] E. Ozcan, Shape recognition using genetic algorithms, International IEEE Conference on Evolutionary Computation, pp: 411-416, 1996.
- [155] Y. Yokoo, M. Hagiwara, Human faces detection method using genetic algorithm. IEEE International Conference on Evolutionary Computation. pp: 113-118. 20-22 May 1996. Japan

- [156] K. P. Han, K.W. Song, E.Y. Chung, S.J. Cho, Y.H. Ha, Stereo Matching using genetic algorithm with adaptive chromosomes, *Pattern recognition*, Vol°34, pp: 1729-1740, 2001.
- [157] N. Lassouaoui, L. Hamami, Genetic algorithms and multifractal segmentation of cervical cell images, *Seventh International Symposium on Signal Processing and its Applications ISSPA 2003*, Vol°2, pp:1-4, 1-4 Juillet 2003, Paris, FRANCE.
- [158] N. Lassouaoui, L. Hamami, Classification of the Cervical Cell Images by Labelling, *Arab Conference on Information Technology ACIT'2003*, Vol°2, pp:682-693, 20-23 Déc. 2003, Alexandrie, Egypte.
- [159] N. Lassouaoui, L. Hamami, Réalisation d'une base de données pour le module de reconnaissance du système de dépistage du cancer du col utérin, *Conférence Internationale de IEEE : Sciences Electroniques, Technologies de l'Information et des Télécommunications SETIT 2003*, 17 -21 Mars 2003, Sousse, Tunisie.
- [160] Abdenour Yaker, *Cancérologie Générale, Anatomie Pathologique*, Office des Publications Universitaires OPU, 1985.