

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE BLIDA

INSTITUT DES SCIENCES EXACTES
DEPARTEMENT DE MATHEMATIQUES

Mémoire de Magister

EN
MATHEMATIQUES APPLIQUEES

OPTION
MODELISATION MATHEMATIQUE
ET TECHNIQUES DE DECISION

THEME:

ANALYSE COMPARATIVE DES METHODES DE
SEGMENTATION DES IMAGES A NIVEAU DE GRIS

APPLICATION AUX IMAGES SATELLITAIRES

Par MOKHTARI Djamila

Soutenu publiquement le : 19 Décembre 1998

Devant le jury composé de:

Président	:	Mr M. DJEDDI	M.C. (U. Boumerdès)
Rapporteurs	:	Mr A. AISSANI	PROF. (USTB)
	:	Mr N.I. BACHARI	C.C. (USTO)
Examineurs	:	Mr GUESSOUM	M.C. (USTB)
	:	Mr S. LARABI	C.C. (USTHB)
	:	Mme N. BENBLIDIA	C.C.(USTB)



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

« وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ »

صَدَقَ اللَّهُ الْعَظِيمُ

AVANT PROPOS

Qu'il me soit permis d'exprimer ma profonde gratitude à *DIEU* qui m'a offert la possibilité d'arriver au terme de ce travail.

Je voudrais aussi remercier très sincèrement, tous ceux qui de près ou de loin ont contribué à l'élaboration de ce modeste travail. Je souhaiterais particulièrement faire part de ma gratitude à :

Monsieur A. AISSANI, Professeur à l'U.S.T.Blida, qui n'a pas lésiné sur son temps pour m'apporter les orientations et corrections nécessaires à la réalisation de cette thèse.

Monsieur N.I. BACHARI, Chargé de cours à l'U.S.T.Oran, qui m'a prodigué des conseils et des orientations dans des périodes de doute.

Messieurs et Mesdames les membres du jury, qui malgré leurs occupations m'ont fait l'honneur d'accepter de juger ce travail.

Madame N. VINCENT, Professeur à l'E3I, Université de Tours, qui par ses orientations, m'a été d'un concours inappréciable, dans l'élaboration de la troisième partie.

Monsieur F. BENHAMOUDA du C.N.T.S d'Arzew, pour m'avoir fourni une documentation précieuse.

Tout le personnel de la maison d'édition HOUMA de Bab El Oued, qui m'ont très souvent épargnée du trajet dangereux (Alger- Oran).

Monsieur A. ABDELLAOUI de l'I.S.T de l'USTHB, qui m'a remis les images satellites nécessaires à mon application.

Tout le personnel de la bibliothèque de l'I.S.E de l'USTB, pour leur compréhension et leur patience.

Mes amis de l'E.N.S.H. et du Département de Mathématiques de l'I.S.E. de l'USTBlida, qui par leurs encouragements, m'ont quelques fois aidée mais beaucoup aidée.

Aux membres de ma famille, qu'ils me pardonnent de les citer en dernier, et tout particulièrement ma mère et mon frère, qui par leur soutien moral permanent ont pu créer un climat favorable à la réalisation de ce travail.

Résumé

L'analyse ou l'interprétation des images a pour but essentiel la reconnaissance d'objets ou la détection des zones homogènes. Les images satellitaires sont caractérisées par l'existence des objets minces dont l'essentiel de l'information se retrouve concentrée dans les segments de plus forte luminance. La détection des segments qui séparent les zones homogènes peut être construite indépendamment de toute application de façon à être intégrés à la boîte outil d'un système général automatique de vision susceptible d'oeuvrer dans différents contextes.

La segmentation constitue une étape cruciale dans la constitution d'une chaîne de traitement d'images satellitaires. Elle consiste à réaliser un partitionnement de l'image en zones homogènes et disjointes. Dans ce travail, nous développons un ensemble de techniques de segmentation que nous appliquons sur des images SPOTXS de l'Atlas Blidéen (zone hétérogène).

Mots clés:

Image- Télédétection- Segmentation- Contours- Régions- Filtrage- Seuillage- Morphologie mathématique- Prétopologie mathématique- Recuit simulé- Générateur de Gibbs- Champs de Markov- Réseaux de Neurones- Modèle de Kohonen.

SOMMAIRE

Résumé.....	i
Sommaire.....	ii
Listes des figures.....	vii
Listes des tableaux.....	viii
Listes des algorithmes.....	viii
Listes des équations.....	ix
INTRODUCTION GENERALE.....	1
CHAPITRE 1: IMAGERIE SATELLITAIRE	
1- La Télédétection.....	4
1-1- Principe de la télédétection.....	4
1-2- Les images satellitaires.....	5
2- Traitement des images satellitaires.....	5
2-1- Numérisation et codage des images.....	5
2-2- Organisation et stockage des données.....	6
3- Représentation des images.....	7
3-1- Histogramme d'une image.....	7
3-2- Perspective d'une image.....	8
3-3- Profil d'une image.....	8
3-4- Isolation des niveaux de gris.....	9
4- Opérations sur les images.....	9
4-1- Opérations ponctuelles.....	9
4-2- Opérations bipoints.....	9
4-3- Opérations localisées.....	10
5- Prétraitement des images.....	10
5-1- Transformations d'histogramme.....	11
5-1-1- Négatif d'une image.....	12
5-1-2- Recadrage de dynamique.....	12
6- Problème d'identification des objets de l'image.....	16
CHAPITRE 2: TAXINOMIE DES METHODES DE SEGMENTATION D'IMAGES: ALGORITHMES ET APPLICATIONS	
Introduction.....	17
Méthodes de segmentation.....	17
1- SEGMENTATION PAR EXTRACTION DE CONTOURS.....	18
1-1- Détection de contours par les méthodes différentielles.....	18
1-1-1- Approche du gradient.....	19
1) Calcul du gradient.....	19
2) Calcul du module du gradient.....	20
3) Analyse du gradient de l'image et recherche des maxima locaux.....	20
1-1-2- Approche du laplacien.....	21
1) Calcul du laplacien.....	21
2) Passage par zéro.....	21
3) Seuillage.....	21
1-2- Détection de contours par les méthodes de filtrage.....	22
1-2-1- Méthodes de filtrage linéaire.....	23
1) Filtre Passe bas.....	23
2) Filtre passe haut.....	23
3) Masque de Robert.....	23
4) Masques de Prewitt.....	23
5) Masques de Sobel.....	24

1-2-2- Méthodes de filtrage non linéaire.....	24
1) Filtre de la médiane.....	24
2) Filtres minimum et maximum.....	24
1-2-3- Méthodes de filtrage adapté.....	25
1) Opérateur de Kirsh.....	25
2) Opérateur moyenne isotropique.....	25
1-3- Application des filtres.....	25
1-3-1- Affichage des images.....	26
1-3-2- Affichage des Histogrammes des niveaux de gris.....	27
1-3-3- Analyse comparative.....	29
1-3-4- Conclusion.....	31
2- SEGMENTATION PAR SEUILLAGE AUTOMATIQUE.....	32
2-1- Méthodes de calcul des seuils.....	34
2-1-1- Méthode de coalescence ISODATA.....	34
2-1-1-1- Principe de la méthode.....	34
2-1-1-2- Algorithme.....	35
2-1-2- Calcul des seuils par l'analyse discriminante.....	36
2-1-2-1- Principe de l'analyse discriminante.....	36
2-1-2-2- Algorithme.....	37
2-2- Application des méthodes du Seuillage.....	38
2-2-1- Affichage des Images et des Histogrammes.....	38
2-2-2- Analyse comparative des résultats.....	40
2-2-3- Conclusion.....	41
2-3- Seuillage local.....	42
3- SEGMENTATION PAR REGIONS.....	43
3-1- Méthodes globales ou par séparation.....	43
3-1-1- Principe de la séparation.....	43
3-1-2- Algorithme de la séparation des régions.....	44
3-1-3- Exemple d'application.....	44
3-2- Méthodes locales ou par fusion.....	46
3-2-1- Principe de la fusion.....	46
3-2-2- Algorithme de la fusion.....	46
3-2-3- Exemple d'application.....	47
3-3- Méthodes par séparation/fusion.....	47
3-4- Application de la Segmentation par régions.....	48
3-4-1- Images et Histogrammes des niveaux de gris.....	48
3-4-2- Analyse comparative.....	49
3-4-3- Conclusion.....	50
4- METHODES ALGEBRIQUES.....	51
4-1- La morphologie mathématique.....	51
4-1-1- Dilatation.....	51
4-1-2- Erosion.....	52
4-1-3- Squelettisation.....	53
4-2- Application de la morphologie mathématique.....	53
4-2-1- Affichage des images.....	54
4-2-2- Analyse comparative.....	54
4-2-3- Conclusion.....	55
4-3- La prétopologie mathématique.....	55
4-3-1- Généralités sur les structures prétopologiques.....	55
4-3-2- Description de la méthode.....	57
4-3-3- Principe de la méthode.....	57

4-3-4- Algorithme de la segmentation par prétopologie.....	58
4-4- Application de la prétopologie mathématique.....	59
4-4-1- Images et Histogrammes des niveaux de gris.....	59
4-4-2- Analyse des résultats.....	59
4-4-3- Discussion des résultats.....	60
4-4-4- Procédure de sélection de contours.....	60
4-4-5- Conclusion.....	61

CHAPITRE 3: APPROCHE ALEATOIRE DU RECUIT SIMULE EN SEGMENTATION DES IMAGES

INTRODUCTION	62
1- PRESENTATION DE L'APPROCHE	62
1-1- Le recuit thermique.....	62
1-2- Le recuit simulé	64
1-2-1- Aspects algorithmiques du recuit simulé	65
1-2-2- Aspects mathématiques du recuit simulé	66
1-2-2-1- Paramètres du recuit simulé.....	67
1-2-3- Utilisation du recuit simulé.....	68
2- LE RECUIT SIMULE EN SEGMENTATION DES IMAGES:	69
2-1- Modèle de segmentation « Recuit1 ».....	69
2-1-1- Principe de la méthode	69
2-1-2- Aspects théoriques de l'algorithme.....	70
2-1-2-1- Modélisation.....	70
2-1-2-2- Paramètres de Recuit1.....	70
2-1-2-3- Opération de seuillage	73
2-1-2-4- Algorithme Recuit1.....	74
2-1-3- Application de Recuit1.....	74
2-1-3-1- Affichage des résultats.....	74
2-1-3-2- Interprétation des résultats de l'algorithme Recuit1	76
2-2- Modèle de segmentation: Recuit2.....	77
2-2-1- Principe de la méthode	77
2-2-2- Aspects théoriques de l'algorithme.....	77
2-2-2-1- Modélisation.....	77
2-2-2-2- Paramètres de Recuit2.....	77
2-2-2-3- Algorithme Recuit2.....	80
2-2-3- Application de Recuit2.....	80
2-2-3-1- Affichage des résultats.....	81
2-2-3-2- Interprétation des résultats de l'algorithme Recuit2	82
2-3- Recuit simulé avec champs de Markov (Recuit3).....	83
2-3-1- Généralités sur les champs de Markov.....	83
2-3-1-1- Définitions d'un champ aléatoire.....	83
2-3-1-2- Systèmes de voisinage et cliques.....	84
2-3-1-3- Définition d'un champ de Markov.....	84
2-3-1-4- Potentiel de Gibbs et champ de Gibbs.....	85
2-3-2- Aspects théoriques de l'algorithme Recuit3.....	85
2-3-2-1- Modélisation locale du champ X.....	86
2-3-2-2- Hypothèse de stationnarité.....	86
2-3-2-3- Représentation du champ de Markov X par une distribution de Gibbs.....	87
2-3-2-4- Simulation du champ X.....	88

2-3-2-5- Recherche de la configuration optimale du champ X.....	89
2-3-2-6- Algorithme Recuit3.....	89
2-3-3- Application de Recuit3.....	90
2-3-3-1- Affichage des résultats.....	90
2-3-3-2- Interprétation des résultats de l'algorithme Recuit3.....	91
2-4- Comparaison des résultats des 3 modèles.....	92

CHAPITRE 4: APPROCHE CONNEXIONISTE EN SEGMENTATION D'IMAGES

1- GENERALITES SUR LES RESEAUX DE NEURONES ARTIFICIELS.....	93
1-1- Caractéristiques des réseaux de neurones artificiels.....	94
1-1-1- Les unités de calcul.....	94
1-1-2- Architecture des réseaux.....	96
1-1-3- Protocole de communication.....	97
1-1-3-1- Mécanisme d'apprentissage	97
1-1-3-2- Règle de Hebb.....	97
2- MODELE DE KOHONEN.....	98
2-1- Types de neurones du réseau de Kohonen.....	99
2-2- Architecture du réseau de Kohonen.....	99
2-3- Règle d'apprentissage du modèle de Kohonen.....	100
3- MODELE DE KOHONEN EN SEGMENTATION D'IMAGES.....	101
3-1- Présentation du réseau.....	101
3-1-1- Structure du réseau	101
3-1-2- Phase d'apprentissage.....	102
3-1-3- Phase de reconnaissance.....	103
3-2- Application et résultats.....	104
3-2-1- Interprétation des résultats.....	105

CHAPITRE 5: PRESENTATION DU LOGICIEL ET DISCUSSION DES RESULTATS

1- ENVIRONNEMENT DE L'EXPERIMENTATION.....	107
1-1- Configuration de la machine.....	107
1-2- Langage de Programmation.....	107
1-3- Description des données de l'expérimentation.....	107
1-4- Structures de données.....	107
2- PRESENTATION DU LOGICIEL DEVELOPPE.....	108
2-1- Description des unités.....	109
2-2- Organigramme du logiciel.....	110
2-3- Opérations de Prétraitement.....	111
2-3-1- Transformations ponctuelles.....	111
2-3-2- Filtrage numérique.....	112
2-4- Les opérations de segmentation.....	112
2-4-1- Détection de contours.....	112
2-4-2- Seuillage.....	112
2-4-3- Segmentation par régions.....	112
2-4-4- Opérateurs algébriques.....	112

2-4-5- Recuit simulé.....	112
2-4-6- Réseau de Kohonen.....	112
3- DISCUSSION DES RESULTATS.....	113
3-1- Conditions de l'expérimentation.....	113
3-2- Analyse des espaces mémoire requis.....	113
3-3- Analyse des temps d'exécution.....	114
3-4- Analyse technique des résultats.....	115
3-5- Récapitulation de la discusion.....	116
CONCLUSION GENERALE.....	118
BIBLIOGRAPHIE.....	120

LISTE DES FIGURES

N°	Titre	Page
1.1	Principe de la télédétection.....	4
1.2	Histogramme à une dimension.....	7
1.3	Image en perspective.....	8
1.4	Profil d'une image.....	8
1.5	Représentation de la transformation « négatif » de l'image.....	12
1.6	Principe du recadrage de dynamique.....	13
1.7	Application du recadrage linéaire à une image.....	13
1.8	Représentation de la transformation logarithmique de l'image.....	14
1.9	Transformation logarithmique d'une image.....	14
1.10	Représentation de la transformation exponentielle de l'image.....	15
1.11	Transformation exponentielle d'une image.....	15
2.1	Représentation d'un contour.....	18
2.2	Dérivation en présence d'un contour.....	22
2.3	Application des filtres sur une image satellitaire.....	27
2.4	Histogrammes des images traitées par les filtres.....	29
2.5	Histogramme bimodal.....	32
2.6	Histogramme multimodal.....	33
2.7	Contribution des histogrammes de niveaux de gris.....	33
2.8	Application du seuillage.....	39
2.9	Application du seuillage local.....	42
2.10	Principe de la segmentation par régions.....	43
2.11	Opération de division d'une image.....	45
2.12	Représentation du résultat de la division par un arbre quaternaire.....	45
2.13	Graphe de contiguïté des régions.....	47
2.14	Première itération de la fusion. Graphe de contiguïté.....	47
2.15	Application de la segmentation par régions.....	49
2.16	Types d'éléments structurants.....	51
2.17	Application d'un élément structurant.....	51
2.18	Application de la dilatation.....	52
2.19	Application de l'érosion.....	53
2.20	Application de la squelettisation.....	53
2.21	Application des opérateurs morphologiques.....	54
2.22	Application de l'algorithme 2.5.....	59
2.23	Elimination des contours négligeables.....	61
3.1	Phénomène de refroidissement d'un système.....	62
3.2	Paysage d'énergie.....	63
3.3	Contribution des histogrammes / Fragmentation.....	69
3.4	Application de Recuit1: Images et histogrammes.....	75
3.5	Application de Recuit1- Variations des température et énergie.....	75
3.6	Application de Recuit2-Images.....	81
3.7	Application de Recuit2- Variations des température et énergie.....	82
3.8	Application de Recuit3-Images et Histogrammes.....	91
4.1	Schéma d'un R.N.A.....	93
4.2	Schéma d'un neurone artificiel.....	95
4.3	Structure d'un R.N.A.....	96
4.4	Règle de Hebb.....	97
4.5	Phénomène de réponse localisée.....	98

4.6	Interaction latérale entre neurones.....	98
4.7	Structure du réseau de Kohonen.....	99
4.8	Connexions du R.N.A.....	102
4.9	Résultats de l'application.....	104
4.10	Résultats de l'application.....	105
4.11	Résultats de l'application.....	105
5.1	Organigramme du logiciel.....	110
5.2	Variation de l'espace mémoire requis par les méthodes.....	113
5.3	Variation du temps d'exécution des méthodes.....	114

LISTE DES TABLEAUX

N°	Titre	Page
2.1	Application des filtres - Analyse visuelle.....	29
2.2	Application des filtres - Analyse des histogrammes.....	30
2.3	Application des filtres - Temps d'exécution et espace mémoire.....	30
2.4	Application du seuillage - Analyse visuelle.....	40
2.5	Application du seuillage- Paramètres statistiques des images.....	40
2.6	Application du seuillage - Analyse des histogrammes.....	41
2.7	Application du seuillage - Temps d'exécution et espace mémoire.....	41
2.8	Déroulement de l'opération de division.....	45
2.9	Application de la segmentation par régions- Analyse visuelle.....	49
2.10	Application de la segmentation par régions - Analyse des histogrammes.....	50
2.11	Application de la segmentation par régions - Temps d'exécution et espace mémoire...	50
2.12	Application des opérateurs morphologiques - Analyse visuelle.....	54
2.13	Application des opérateurs morphologiques -Temps d'exécution et espace mémoire...	55
2.14	Application de la prétopologie mathématique - Analyse visuelle.....	59
2.15	Application de la prétopologie mathématique - Analyse de l'histogramme.....	60
2.16	Application de la prétopologie mathématique-Temps d'exécution et espace mémoire...	60
3.1	Résultat de l'application de Recuit 1.....	76
3.2	Résultat de l'application de Recuit 2.....	82
3.3	Résultat de l'application de Recuit 3.....	91
3.4	Tableau comparatif des résultats du recuit simulé.....	92
5.1	Avantages et Inconvénients des méthodes de segmentation.....	117

LISTE DES ALGORITHMES

N°	Titre	Page
2.1	Isodata.....	35
2.2	Analyse discriminante: cas de deux classes.....	37
2.3	Séparation de régions.....	44
2.4	Fusion de régions.....	46
2.5	Segmentation par prétopologie mathématique.....	58
3.1	Recuit simulé.....	66
3.2	Hist_Energie.....	71
3.3	Hist_Generer.....	72
3.4	Seuillage.....	73
3.5	Recuit 1.....	74
3.6	Img_Energie.....	78
3.7	Img_Perturber.....	78

3.8	Img_Generer.....	79
3.9	Recuit2.....	80
3.10	Recuit simulé avec échantillonneur de Gibbs.....	89
3.11	Recuit3.....	89
4.1	Algorithme d'apprentissage de Kohonen.....	100
4.2	Phase d'apprentissage	103
4.3	Phase de reconnaissance.....	104

LISTE DES EQUATIONS

N°		Page
1.1	11
1.2	12
1.3	12
1.4	14
1.5	15
2.1	19
2.2	19
2.3	19
2.4	20
2.5	20
2.6	20
2.7	20
2.8	20
2.9	21
2.10	21
2.11	24
2.12	24
2.13	34
2.14	34
2.15	35
2.16	35
2.17	36
2.18	36
2.19	36
2.20	36
2.21	36
2.22	46
2.23	46
2.24	60
3.1	63
3.2	64
3.3	65
3.4	66
3.5	66
3.6	67
3.7	68
3.8	68
3.9	71
3.10	78

3.11	84
3.12	85
3.13	85
3.14	86
3.15	87
3.16	87
3.17	87
3.18	87
3.19	87
3.20	88
3.21	88
3.22	88
4.1	100
4.2	100
4.3	102

INTRODUCTION GENERALE

INTRODUCTION GENERALE

La gestion du milieu, dans les pays en développement fait l'objet d'un intérêt particulier en cette fin de siècle. Elle constitue une priorité et nécessite la disponibilité de données variées et spécifiques à chaque domaine d'application, notamment en agriculture, géologie, urbanisme, hydraulique, etc....

Ces données peuvent provenir de prélèvements, de mesures ou d'observations effectuées in situ, ou bien d'images prises à distance depuis une plate forme aéroportée (hélicoptère, avion, ballon) ou spatiale (satellite), d'où l'avènement de la *téledétection*, intégrant les développements les plus récents de la recherche spatiale, de la physique et de l'informatique.

La télédétection est une technique de repérage d'objets ou de phénomènes à distance, sans contact physique. Son but est d'arriver à *identifier* les objets observés, afin de prendre les décisions économiques, politiques ou stratégiques conséquentes.

Dans l'identification des objets de l'image, on intègre des informations sémantiques, sur toute l'image à analyser en se servant à la fois des *données* de l'image, ainsi que d'une base de connaissances appropriées utilisant les techniques de l'intelligence artificielle. C'est le traitement *haut niveau*.

Cependant, les données contenues dans les images, en provenance des sondes spatiales et autres satellites ne peuvent être exploitées directement, dans l'identification. Elles nécessitent au préalable, des traitements numériques sophistiqués, dits de *bas niveau*. Parmi ces traitements, on distingue:

- La numérisation et le codage des images,
- l'organisation et le stockage des images,
- le prétraitement des images et
- la *segmentation* des images.

Les traitements bas niveau, s'appliquent sur des données numériques, sans tenir compte de l'aspect contextuel de l'image.

Les trois premières phases sont décrites brièvement, dans le premier chapitre de cette thèse. Cependant, l'étape de la segmentation de l'image qui représente l'objectif de notre travail, constitue la majeure partie de ce document.

La segmentation est une étape cruciale, dans le traitement des images. Elle permet de grouper les petits blocs (pixels) produits lors de la numérisation, dans des blocs plus importants appelés *segments* correspondants généralement à des traits significatifs des objets dans l'image, on parle alors de *contours*.

Quelques fois, ces segments correspondent à l'intérieur de l'objet lui même plutôt qu'à ses frontières, il s'agit dans ce cas d'une segmentation par *régions*. Dans le premier cas, on utilise des méthodes basées sur les contours. Dans le deuxième cas, on utilise des méthodes basées sur les régions homogènes.

La segmentation des images permet la répartition de l'image en classes homogènes selon un ou plusieurs critères donnés (Niveau de gris, couleur, texture,...). Dans le cadre de cette étude, nous nous limitons à la segmentation des images selon le critère niveau de gris.

La segmentation des images à niveaux de gris se ramène à un problème de classification. Le programme traitant doit pouvoir décider de l'appartenance d'un pixel à une classe ou à une autre, en lui attribuant une nouvelle valeur de niveau de gris la caractérisant. Ces classes peuvent être représentées par les frontières des objets de l'image -contours- ou par l'intérieur de ces objets -régions-.

Le but du travail est de présenter différentes méthodes mathématiques et algorithmiques de segmentation des images à niveaux de gris et de les synthétiser en une étude comparative. Chacune de ces méthodes est caractérisée par des outils et une méthodologie qui lui sont propres.

Afin d'atteindre cet objectif, cette thèse a été scindée en cinq chapitres, le premier est consacré à une synthèse bibliographique se rapportant à l'imagerie satellitaire.

Le second chapitre concerne le développement mathématique et algorithmique ainsi qu'une étude comparative de différentes méthodes de segmentation d'images. La liste des méthodes présentées n'est pas exhaustive, vu la diversité de celles-ci et les nombreux écrits à ce propos. Les méthodes développées sont:

- Les méthodes de détection de contours par le filtrage numérique.
- Les méthodes de seuillage basées sur des paramètres statistiques de l'image tels que l'histogramme.
- La méthode récursive de la segmentation par régions.
- Les méthodes algébriques basées sur la morphologie mathématique et la prétopologie mathématique.

Dans le troisième chapitre, nous nous proposons d'utiliser l'approche aléatoire du recuit simulé en segmentation d'images. L'affectation des éléments de l'image à une classe se fait selon une certaine probabilité. Dans ce volet, nous présentons trois algorithmes de segmentation d'images basés sur le principe du recuit simulé. Le premier

modèle est basé sur des transformations effectuées sur l'histogramme de l'image, le second est basé sur des transformations globales de l'image et enfin dans le dernier algorithme, nous introduisons les champs de Markov afin de traiter l'image d'une manière locale.

Quant au chapitre quatre, il est consacré à l'approche connexioniste en segmentation d'images. Nous y proposons un algorithme d'apprentissage de détection de contours basé sur le modèle de Kohonen. Nous exploitons une propriété de ce modèle qu'est la non-supervision, afin de mettre au point notre algorithme. L'algorithme proposé n'apprend pas au réseau des formes déterminées mais lui permet, plutôt, d'acquérir un pouvoir séparateur afin de distinguer les contours de l'intérieur des régions.

Dans le cinquième chapitre, nous présentons le logiciel développé, regroupant les différentes méthodes citées ci-dessus, ainsi, qu'une discussion sur les résultats obtenus par ces différentes méthodes.

Enfin, nous clôturons cette étude par une conclusion. Nous y présentons aussi quelques perspectives relatives à ce travail.

CHAPITRE 1 :

IMAGERIE SATELLITAIRE

- 1- La Télédétection.
- 2- Traitement des Images Satellitaires.
- 3- Représentation des Images.
- 4- Opérations sur les Images.
- 5- Prétraitement des Images.
- 6- Problème d'Identification des Objets de l'Image.

Chapitre 1: **IMAGERIE SATELLITAIRE**

1- La Télédétection:

La télédétection est une technique de repérage d'objets ou de phénomènes à distance, sans contact physique. Historiquement, elle prend sa source de la photographie aérienne, utilisée depuis 1954, en sciences de la terre ainsi que dans les missions exploratoires. Les photographies aériennes, bien qu'elles possèdent un excellent pouvoir séparateur et permettent de distinguer les détails, présentent un inconvénient majeur qui est le manque de recul pour examiner les objets ou les phénomènes de grande extension géographique.

En 1965, les astronautes américains, embarqués à bord de la capsule spatiale Appolo, ont pu photographier la terre de plus loin. Ces images spatiales marquèrent les prémises de l'avènement de la télédétection spatiale. Quelques années plus tard, en 1972, les américains ont lancé le premier satellite LANDSAT 1 destiné à l'inventaire des ressources terrestres. Depuis, plusieurs satellites de télédétection ont été lancés, stimulant à chaque fois un renouveau dans le développement des techniques de la télédétection [13].

1-1- Principe de la Télédétection:

La télédétection utilise des **satellites** à basse altitude, ou géostationnaires gravitant autour de la terre et effectuant des prises de vue périodiques d'une région déterminée du globe. Les images obtenues seront transmises vers des **capteurs paraboliques** au niveau des stations terriennes, puis stockées sur **ordinateur**, pour un traitement ultérieur, afin d'identifier les objets de la terre [5].

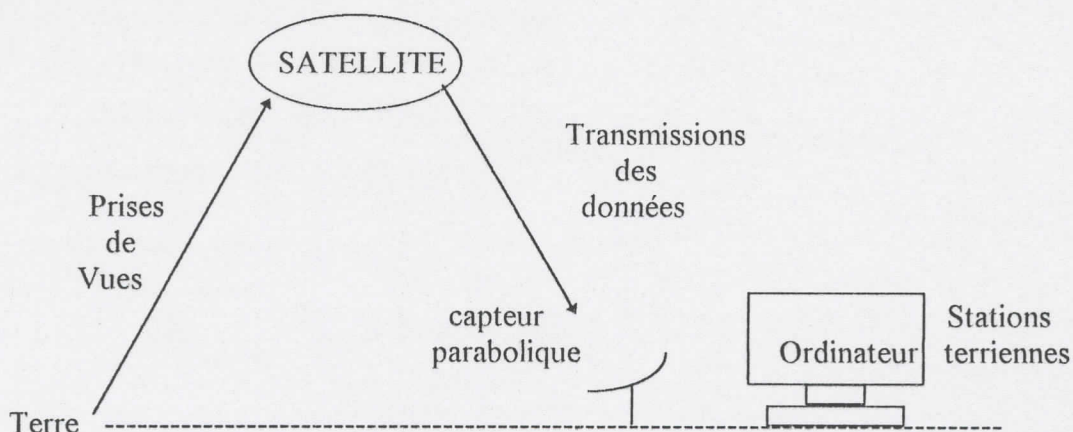


Figure 1.1 : Principe de la télédétection.

1-2- Les Images Satellitaires [5],[39]:

Parmi les satellites de télédétection, il existe:

- Le LANDSAT MSS (MultiSpectral Scanner),
- le LANDSAT TM (Thématique Mapper) et
- le système SPOT (Satellite Probatoire pour l'Observation de la Terre).

Pour des raisons de disponibilité des images, nous nous sommes limités dans cette étude, aux images SPOT. Le satellite SPOT est héliosynchrone, il est placé en orbite polaire à 832 Km, balayant la totalité du globe terrestre en un cycle de 26 jours. Il comprend deux capteurs HRV1 et HRV2 (Haute Résolution Visible) identiques mais gouvernés indépendamment.

Chacun de ces deux capteurs effectuent des prises de vue selon deux modes:

- Le mode panchromatique (image en noir et blanc) avec une résolution spatiale de 10x10m. L'étendue de la bande spectrale est de: 0,50 μm à 0,73 μm .
- Le mode multispectral avec une résolution de 20x20m.
 - Bande verte : 0,50 μm à 0,59 μm
 - Bande rouge: 0,60 μm à 0,68 μm
 - Proche Infrarouge: 0,79 μm à 0,89 μm

Chacune de ces trois bandes du spectre caractérise un canal du capteur, respectivement le canal XS1, XS2 et XS3.

2- Traitement des Images Satellitaires:

Les images en provenance des sondes spatiales et autres satellites nécessitent des traitements numériques sophistiqués pour être exploitées dans différents domaines tels que l'urbanisme, les inventaires agricoles, l'exploration spatiale, l'océanographie, la géologie,...

Le but de la télédétection étant l'identification des objets de la terre, il est nécessaire que les images satellitaires subissent des traitements numériques. Parmi ces traitements, la *segmentation* des images est l'étape la plus importante. Néanmoins, d'autres étapes sont indispensables telles que:

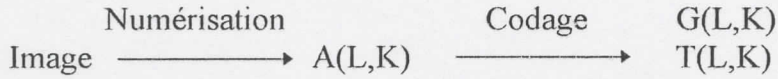
- Numérisation et codage des images.
- Organisation et stockage des images.
- Prétraitement des images.

2-1- Numérisation et Codage des Images [43]:

La numérisation consiste à partager l'image acquise par le capteur du satellite, en un nombre d'éléments appelés pixels, pour donner un tableau bidimensionnel A, à L lignes et K colonnes.

On attribue à chaque pixel $A[i,j]$ une fonction image $G[i,j]$, dite **brillance** proportionnelle à son intensité lumineuse. Cette grandeur, est généralement codée sur huit bits, elle prend donc des valeurs dans la gamme numérique [0,255]. Elle peut être soit une valeur dans

l'échelle des niveaux de gris (images en noir et blanc) ou une nuance de couleur (pour les images couleurs). On peut aussi attribuer à chaque pixel $A[i,j]$, d'autres fonctions images telles que la texture, l'orientation, ...

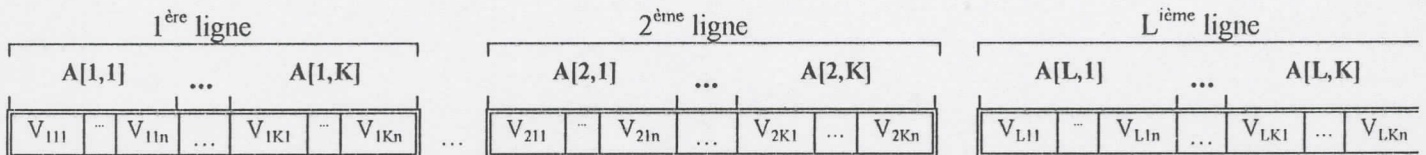


$G(L,K)$: matrice des valeurs de niveaux de gris de l'image.
 $T(L,K)$: matrice des valeurs de textures de l'image.

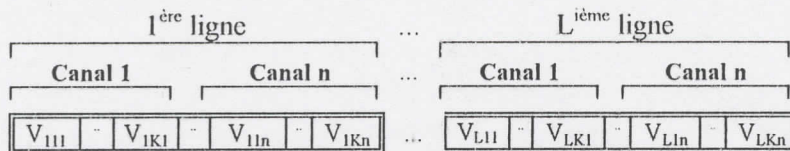
2-2- Organisation et Stockage des Données:

L'image est une matrice à L lignes et K colonnes. Si on considère le mode d'acquisition multispectral des images, on constate que l'image d'une scène est sous forme de n images parallèles, où n est le nombre de canaux du capteur du satellite. Ainsi, pour chaque pixel $A[i,j]$, on attribue un vecteur de n composantes. Mathématiquement, l'image est représentée par une matrice $K \times L \times n$. Pour son stockage plusieurs méthodes sont utilisées [39].

1- **La méthode B.I.P (Band Interleaved by Pixel):** On stocke séquentiellement pour chaque pixel les n valeurs V fournies par les n canaux.



2- **La méthode B.I.L (Band Interleaved by Line):** On stocke séquentiellement, pour chaque ligne de l'image les n valeurs fournies par les n canaux.



3- **La méthode B.S.Q. (Band Séquentiel):** Chaque canal est stocké séparément.

3- Représentation des Images:

Partant de la représentation matricielle de toute l'image, celle-ci peut être représentée par ses paramètres statistiques.

3-1- Histogramme d'une Image [39]:

Avant de traiter une image, il est nécessaire de constituer son histogramme. Celui-ci représente la fréquence d'occurrence de chaque niveau de gris dans l'image. Il est obtenu en analysant la répartition spectrale des valeurs radiométriques des pixels et en comptabilisant ceux qui ont la même luminance sur la totalité de l'image. L'histogramme permet de déterminer le nombre de classes d'objets disponibles sur l'image. Il permet aussi d'analyser entre quelles bornes, est répartie la majorité des niveaux de gris dans le cas d'une image trop claire ou trop foncée.

1- Histogramme à une dimension:

On considère l'image fournie par un seul canal du satellite, la représentation graphique se fait dans un repère bidimensionnel où l'axe des abscisses représente le niveau de gris et l'axe des ordonnées représente le nombre de pixels ayant un même niveau de gris par rapport à l'ensemble des pixels de l'image.

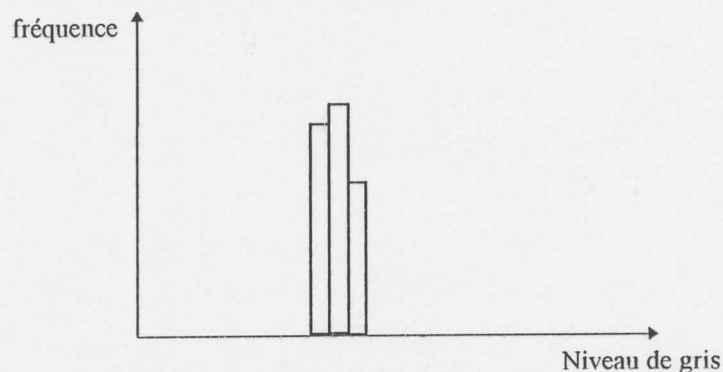


Figure 1.2 : Histogramme à une dimension.

2- Histogramme à deux dimensions:

Dans ce cas de représentation, on considère l'image fournie par 2 canaux. Pour une image SPOT, l'histogramme à 2-D requiert 256x256 niveaux de gris et chaque pixel est représenté par un vecteur à 2 composantes où chaque composante représente le niveau de gris correspondant à un canal.

3- Histogramme à plusieurs dimensions:

Ce type d'histogrammes est utilisé pour des images provenant de satellites à plusieurs canaux.

3-2- Perspective d'une Image:

L'image peut être aussi représentée en perspective; si on considère les trois dimensions: ligne, colonne et niveau de gris [52].

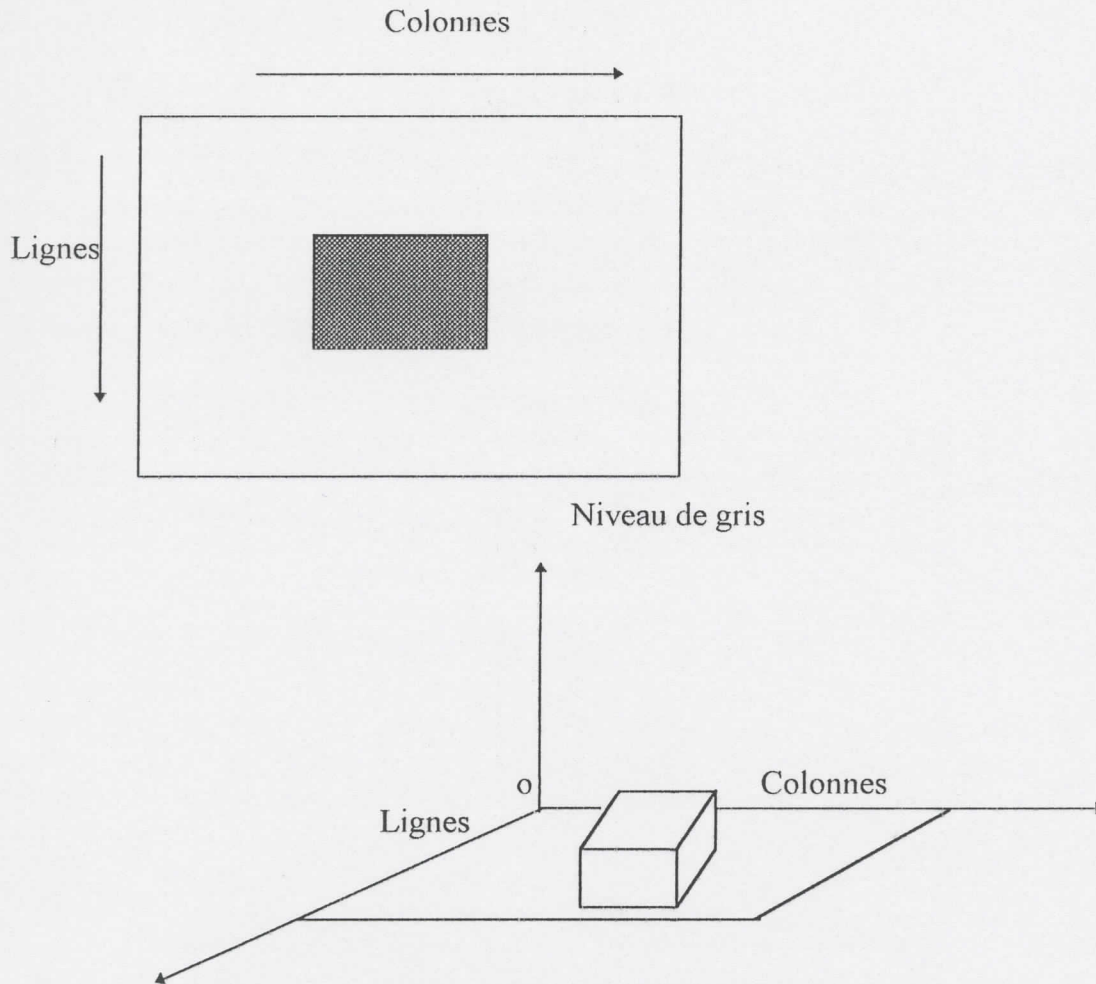


Figure 1.3: Image en perspective.

3-3- Profil d'une Image:

On peut aussi réaliser sur l'image en perspective, une coupe à une ligne bien déterminée de l'image afin d'analyser les variations des niveaux de gris [52].

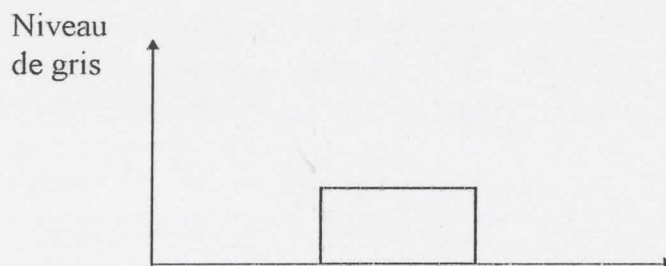


Figure 1.4: Profil d'une image.

3-4- Isolation des Niveaux de Gris:

On peut à partir des coordonnées (i,j) d'un point de l'image, afficher tous les points ayant même niveau de gris. Cela permet de déterminer comment se répartissent dans l'espace les pixels de même intensité [52].

4- Opérations sur les Images [52]:

4-1- Opérations Ponctuelles:

Chaque pixel (i,j) de l'image originale, est transformé en un pixel dont le niveau de gris est obtenu par application d'une certaine fonction f :

$$G'[i,j]=f(G[i,j]) ;$$

ou bien, obtenu à partir d'une table de niveaux de gris préalablement construite en fonction de l'image originale, de son histogramme et du type de traitement souhaité.

4-2- Opérations Bipoints:

Elles utilisent les informations contenues au même emplacement (i,j) , sur deux images A et B , afin de créer une nouvelle image C. Les images A et B représentent une même scène, mais elles sont prises à des instants différents.

1- Addition :

$$C[i,j]= (A[i,j] + B[i,j]) /K$$

K: coefficient de normalisation pour que $C[i,j] \in [0..255]$, généralement, on choisit $K=2$, les valeurs $C[i,j]$ obtenues sont arrondies à leurs valeurs entières supérieures.

2- Soustraction : De la même manière,

$$C[i,j] = K . | A[i,j] - B[i,j]|$$

3- Multiplication : On multiplie la matrice image A par une matrice de correction B, on obtient une matrice C :

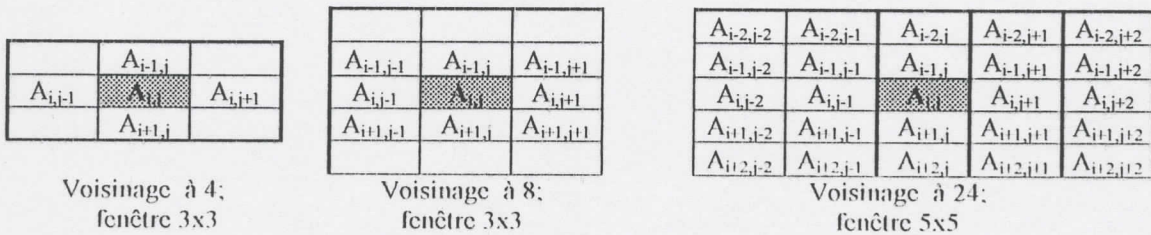
$$C[i,j] = K ((A[i,j]*B[i,j]) + A[i,j])$$

4-3- Opérations Localisées:

Le niveau de gris de chaque pixel de coordonnées (i,j) de l'image initiale, est transformée en fonction des niveaux de gris de ses voisins.

1- Définition du voisinage d'un pixel :

Le voisinage d'un pixel $A[i,j]$ est un ensemble de pixels de l'image, situés à proximité du pixel $A[i,j]$. Il est défini par sa forme et par sa taille.



2- Produit de convolution :

Pour un pixel donné $A[i,j]$, on considère son voisinage et un masque M de même taille, on multiplie le niveau de gris de chaque pixel de la fenêtre avec le facteur de poids correspondant du masque, puis on additionne tous les produits, le résultat obtenu sera la nouvelle valeur de gris du pixel $A[i,j]$.

$$G'[i,j] = \sum_{k=-m}^m \sum_{\ell=-m}^m M[k, \ell] G[i+k, j+\ell]$$

M : masque de taille $2m+1$.

G : matrice image.

G' : matrice image transformée.

5- Prétraitement des Images [39][46]:

L'image est sujette à des dégradations dues aux:

- *Turbulences atmosphériques.*
- *Défauts des systèmes de prise de vue:*
 - Qualité optique de l'objectif du capteur,
 - Bruit, électronique du capteur,
 - Qualité de l'enregistrement sur un support photographique...
- *Défauts de systèmes de numérisation:*
 - Qualité de l'échantillonnage et de la quantification;

En effet, des erreurs de quantification peuvent s'introduire à cause du passage continu-discret.

- *Défauts des systèmes de transmission.*
- *Défauts de systèmes de réception* au niveau des stations terriennes.
- *Défauts de systèmes de visualisation* : qualité du dispositif d'affichage, du point de vue:
 - Résolution,
 - Gamme de couleurs utilisée
 - Etendue de l'intervalle des niveaux de gris utilisé pour représenter l'image.

Pour cela, il est nécessaire que l'image subisse des prétraitements tels que la restauration et l'amélioration.

La restauration consiste en la reconstitution de l'image dégradée en une image nette, aussi proche que possible de l'image avant dégradation. Elle s'appuie principalement sur la détermination du modèle mathématique du processus ayant engendré la dégradation puis l'application du modèle inverse sur l'image [33].

L'amélioration d'une image quant à elle est appliquée sans connaissance préalable du phénomène de dégradation. Elle consiste à générer à partir de l'image initiale une nouvelle image dont l'aspect visuel est amélioré où certains traits caractéristiques sont mis en évidence (renforcement des contours, utilisation de la couleur...). En appliquant différentes techniques d'amélioration notamment :

- Les transformations d'histogramme.
- Le filtrage digital.

5-1- Transformations d'Histogramme[39] [52]:

Nous avons vu que l'histogramme de l'image représente la fréquence d'occurrence de chaque niveau de gris dans l'image. Les transformations d'histogramme visent à améliorer le contraste de l'image. Ce sont des **opérations ponctuelles** agissant sur le niveau de gris de chaque point de l'image indépendamment de ces voisins. Celui là est transformé en un pixel dont le niveau de gris est obtenu par application d'une certaine fonction f:

$$G'[i,j]=f(G[i,j]) \quad (1.1)$$

Ou bien obtenu à partir d'une table de niveaux de gris appelée LUT(Look Up Table) préalablement construite en fonction de l'image originale, de son histogramme et du type de traitement souhaité.

Il existe plusieurs transformations ponctuelles:

5-1-1- Négatif d'une Image: L'image est transformée comme suit :

$$G'[i,j] = 255 - G[i,j] \quad (1.2)$$

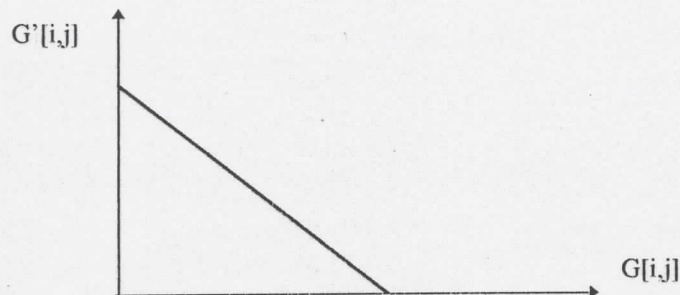


Figure 1.5: Représentation de la transformation « négatif de l'image ».

5-1-2- Recadrage de Dynamique: (Elargissement de contraste)

Le recadrage de dynamique est utilisé dans le cas d'images trop claires (les niveaux de gris sont tassés vers la borne supérieure de l'échelle des niveaux de gris), ou trop foncées (les niveaux de gris sont tassés vers la borne inférieure de l'échelle des niveaux de gris) ou bien peu contrastées (les niveaux de gris des pixels sont regroupés dans un intervalle étroit).

Le but du recadrage est de redistribuer les niveaux de gris des pixels de l'image de façon à utiliser optimalement l'échelle des niveaux de gris disponibles.

1- Elargissement linéaire de contraste:

On choisit un intervalle [Min,Max] de niveaux de gris. Les pixels dont les niveaux de gris sont dans [Min,Max] sont transformés selon l'équation (1.3), ceux dont les niveaux de gris sont inférieur à Min auront une valeur de niveau de gris nulle, les autres seront forcés à un niveau de gris égal à 255.

$$G'[i,j] = \begin{cases} (G[i,j] - \text{Min}) \frac{255}{\text{Max} - \text{Min}} & \text{Si } \text{Min} \leq G[i,j] \leq \text{Max} \\ 0 & \text{Si } G[i,j] < \text{Min} \\ 255 & \text{Si } G[i,j] > \text{Max} \end{cases} \quad (1.3)$$

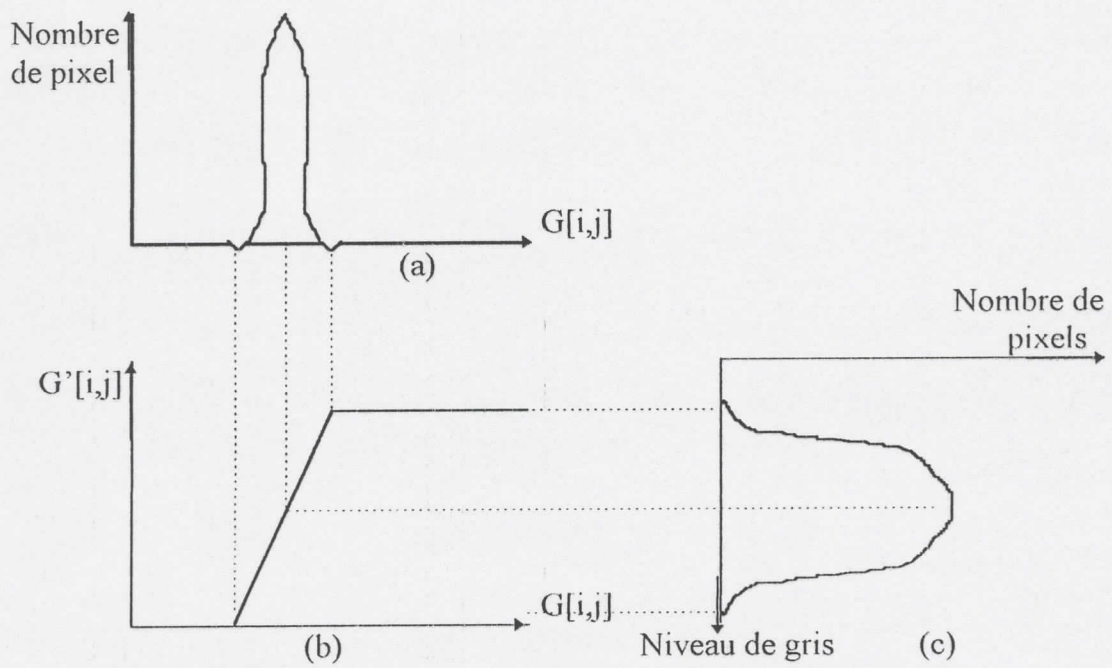
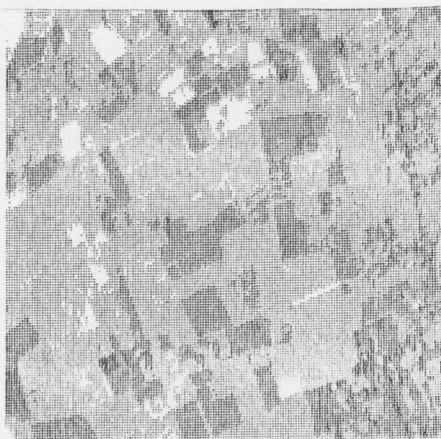


Figure 1.6 : Principe du recadrage de dynamique.

(a) histogramme de l'image originale (b) fonction de recadrage (c) histogramme de l'image recadrée.



(a)



(b)

Figure 1.7: Application du recadrage linéaire à une image.

(a) image originale (b) image recadrée avec [40, 70].

2- Elargissement non linéaire de contraste:

* **Transformation logarithmique:** Elle permet d'éclaircir les parties sombres de l'image :

$$G'[i,j] = K \cdot \text{Log } G[i,j] \quad (1.4)$$

K est un coefficient à déterminer.

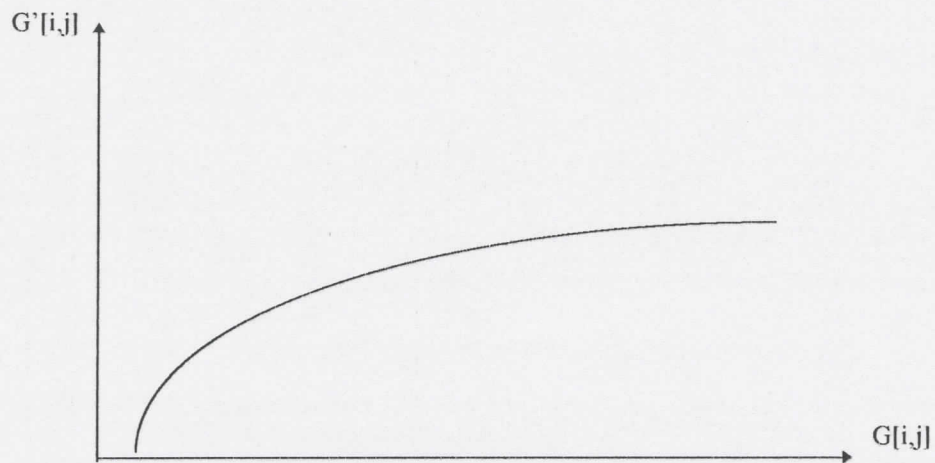
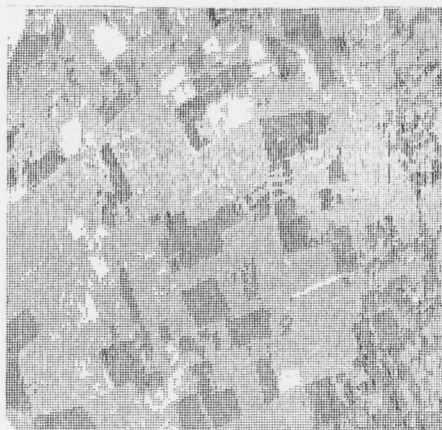


Figure 1.8: Représentation de la transformation logarithmique de l'image.



(a)



(b)

Figure 1.9: Transformation logarithmique d'image.
(a) image originale (b) image transformée avec $K=0,5$

* Transformation exponentielle : Elle permet d'assombrir les parties claires de l'image:

$$G'[i,j] = K \cdot \text{Exp } G[i,j] \quad (1.5)$$

K est un coefficient à déterminer.

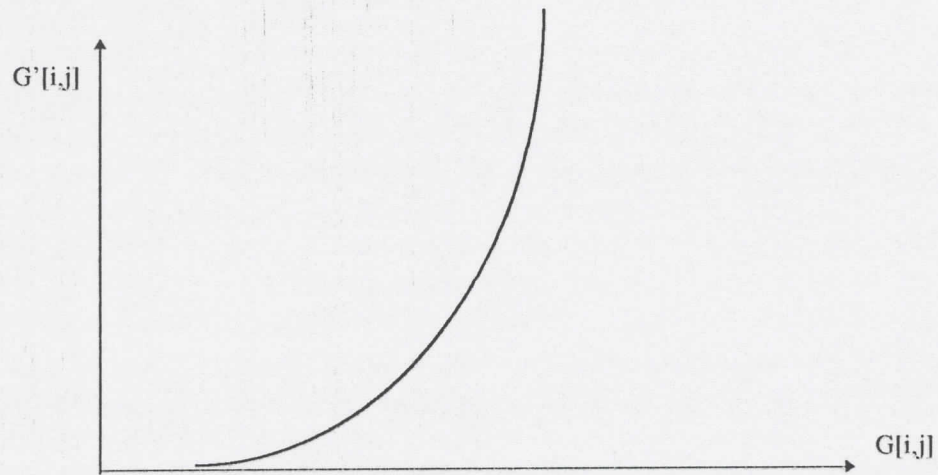
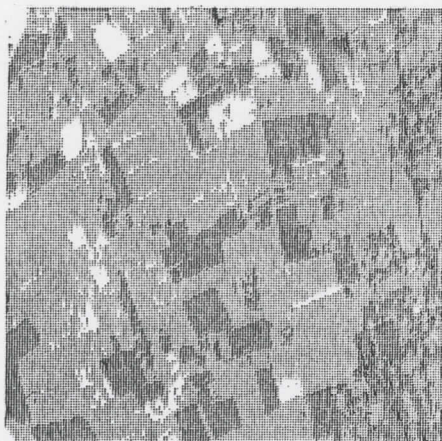
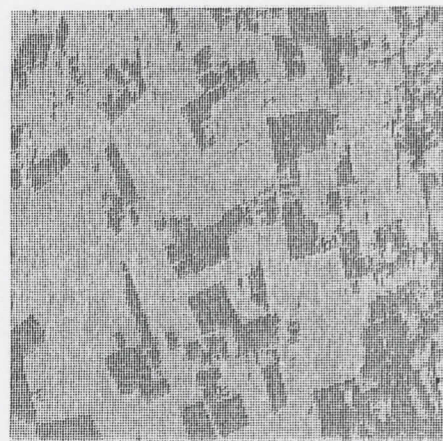


Figure 1.10: Représentation de la transformation exponentielle de l'image.



(a)



(b)

Figure 1.11 : Transformation exponentielle d'image
(a) image originale (b) image transformée avec $K=$

6- Problème d'Identification des Objets dans l'Image:

L'identification des objets de l'image, constitue le but de la télédétection. Elle permet l'intégration des informations sémantiques sur toute l'image à analyser en se servant à la fois des données issues de la *segmentation* ainsi que d'une base de données appropriées utilisant les techniques de l'intelligence artificielle et contenant d'autres informations sur l'image, telles que les coordonnées géographiques de la région représentée, sa couverture géologique....

Les images satellitaires sont caractérisées par l'existence des objets minces dont l'essentiel de l'information se trouve concentré dans les segments de plus forte luminance [6]. La détection des segments qui séparent les zones homogènes peut être construite indépendamment de toute application de façon à être intégrés à un système général automatique de vision susceptible d'oeuvrer dans différents contextes [21].

La segmentation vise une nouvelle représentation de l'image, dans laquelle les points images partageant une même propriété sont groupés pour former des objets homogènes. Ces objets sont décrits par leurs frontières ou par les pixels qui les composent. La *segmentation* est la plus importante des étapes de traitement des images. Elle permet de réduire la quantité gigantesque des informations de l'image, en effectuant une classification des pixels qui la composent.

Nous présentons dans le chapitre suivant, une étude comparative des techniques de segmentation des images.

CHAPITRE 2 :

***TAXINOMIE DES METHODES
DE SEGMENTATION D'IMAGES
ALGORITHMES ET APPLICATIONS***

- 1- Segmentation par Extraction de Contours.
- 2- Segmentation par Seuillage Automatique.
- 3- Segmentation par Régions.
- 4- Segmentation par les Méthodes Algébriques.

Chapitre 2: *TAXINOMIE DES METHODES DE SEGMENTATION D'IMAGES: Algorithmes et Applications*

INTRODUCTION:

La segmentation consiste à partitionner l'espace géométrique de l'image en régions connexes disjointes homogènes. Ce partitionnement est basé sur des critères de similarité et/ou de dissimilarité d'attributs. L'attribut le plus souvent utilisé est le niveau de gris pour des images monochromes et la couleur pour des images multispectrales. D'autres attributs tels que la texture et le mouvement peuvent être utilisés en segmentation [7].

La segmentation des images à niveau de gris se ramène à un problème de classification. Le programme traitant doit pouvoir décider de l'appartenance d'un pixel à une région ou à une autre, en lui attribuant une nouvelle valeur de niveau de gris caractérisant la région. Ces régions peuvent être décrites par des **frontières** ou bien par les **pixels** qui les composent. Dans le premier cas, on utilise des méthodes basées sur les bords ou **contours**. Dans le deuxième cas, on utilise des méthodes basées sur les **régions homogènes**.

METHODES DE SEGMENTATION:

Les méthodes de segmentation des images sont généralement fondées sur deux principes: La recherche des discontinuités locales (détection de contours) ou sur la détection des zones d'attributs homogènes (extraction de régions).

On rencontre dans la littérature plusieurs techniques de segmentation. Elles font généralement, l'objet d'un état de l'art. On peut se reporter aux travaux de Torre et Poggio [51], de Canny [12] et de Nalwa et Binford [41] pour les techniques de détection des contours, de Pratt [44] pour les techniques basées sur les régions, de Weszka et Rosenfield [53][33] pour les techniques de seuillage et les travaux de Emptoz [17], de Haralick [24], de Geman [20] pour d'autres techniques de segmentation en général.

Nous présentons dans ce qui suit, un ensemble non exhaustif de méthodes de segmentation. Ces méthodes sont regroupées selon la méthodologie et les outils utilisés:

- méthodes de détection de contours (méthodes différentielles, le filtrage).
- méthodes basées sur des paramètres statistiques telle que le seuillage et la croissance des régions.
- méthodes algébriques basées sur la structure ensembliste de l'image et utilisant la morphologie mathématique et la prétopologie mathématique.
- méthodes stochastiques , où l'affectation aux classes se fait par l'utilisation de vecteurs probabilistes d'affectation de chaque point de l'image à l'ensemble des classes et où chacun des vecteurs évolue au cours d'itérations et converge vers une classification correcte.
- méthodes basées sur l'approche des réseaux de neurones.

Nous présentons dans ce chapitre les méthodes différentielles, le filtrage, le seuillage, la segmentation par régions, et les méthodes algébriques. L'approche aléatoire et l'approche des réseaux de neurones seront présentées ultérieurement.

1- Segmentation par Extraction de Contours:

Dans la segmentation par extraction de contours, on s'intéresse à la forme d'une région plutôt qu'à son contenu. Sa forme peut être décrite par un ensemble de pixels dont le niveau de gris est nettement différent des niveaux de gris des pixels voisins.

Un contour est une zone de transition séparant deux régions adjacentes ayant des caractéristiques différentes. Pratt [44] définit les contours comme des discontinuités locales de la fonction de gris dans l'image. Mathématiquement, on peut énoncer la définition suivante d'un contour: [40],[49],[41]:

« Soit $g(x)$ le profil de l'intensité, dans une image selon une certaine direction. On dit que l'image présente un contour en $x=a$ si la $n^{\text{ième}}$ dérivée de g en a est discontinue. L'ordre de dérivation le plus faible où l'on observe cette discontinuité détermine le type de contours. On définit une classification des contours selon l'ordre de dérivation. Par exemple, si $g(x)$ est discontinue en a , on est en présence d'un contour en marche d'ordre 0. La discontinuité de $g'(x)$ définit des contours en toit d'ordre 1... »

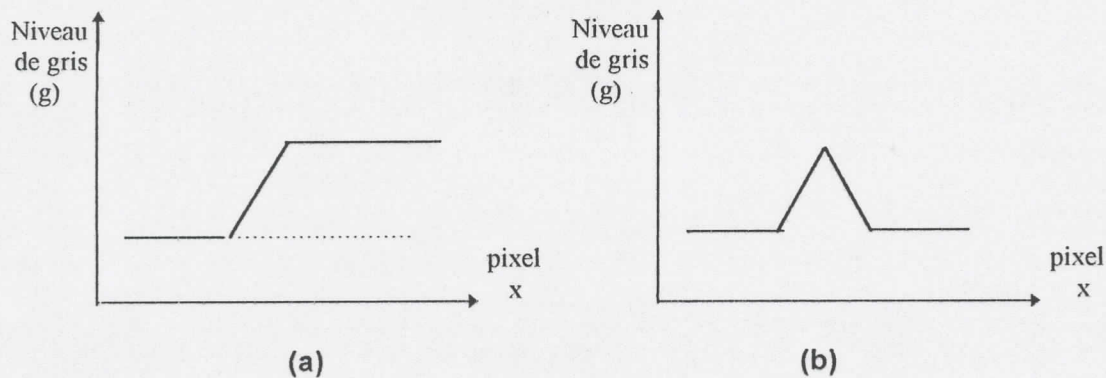


Figure 2.1: Représentation d'un contour.
(a) contour en marche. (b) contour en toit.

Cependant, cette définition nous induit à considérer toute variation de niveau de gris, aussi faible soit elle comme étant un contour d'amplitude plus ou moins grande, ce qui est bien différent du monde réel. Il est, par conséquent, impératif de faire une sélection de contours, ceci est réalisé en fixant un seuil à partir duquel les contours sont éliminés ou gardés [33].

1-1- Détection de Contours par les Méthodes Différentielles:

Soient A : la matrice image (ensemble de pixels).

G : la matrice des niveaux de gris de ces pixels.

Les méthodes différentielles procèdent au calcul du gradient de l'image en chaque point $A[i,j]$, en effectuant une seule dérivation de $G[i,j]$, ou le calcul du laplacien en dérivant deux fois [33],[52].

1-1-1- Approche du Gradient:

Elle comporte trois étapes : le calcul des composantes du vecteur gradient, le calcul de la norme du gradient et le seuillage.

1) Calcul du gradient:

La définition du gradient fait intervenir les deux dérivées premières partielles de la fonction de niveau de gris, selon les directions horizontales et verticales:

$$\text{grad } [i,j] = \left(\overset{\longrightarrow}{\frac{\delta G[i,j]}{\delta i}} ; \overset{\longrightarrow}{\frac{\delta G[i,j]}{\delta j}} \right) \quad (2.1)$$

Cependant la nature discrète de l'image fait que le calcul du gradient par dérivation est impossible, il est nécessaire d'en faire une approximation. Le calcul du gradient dans son aspect le plus simple repose sur un principe fondamental qui est le calcul de la différence des intensités des pixels adjacents.

Pour la détection d'une transition de sens horizontal on a:

$$\text{grad}(i) = \frac{\delta G[i,j]}{\delta i} = \frac{G[i,j] - G[i+1, j]}{(i+1) - i} \quad (2.2)$$

et les transitions verticales son données par:

$$\text{grad}(j) = \frac{\delta G[i,j]}{\delta j} = \frac{G[i,j] - G[i, j+1]}{(j+1) - j} \quad (2.3)$$

Ces deux opérations permettent le calcul du gradient dans deux directions : horizontale (Ouest \rightarrow est) et verticale (Nord \rightarrow Sud).

Il se trouve que dans certains cas, on souhaite calculer le gradient d'un point par rapport à tous ces voisins ie dans les quatre directions (Ouest \rightarrow Est) (Nord \rightarrow Sud) (Sud \rightarrow Nord) (Est \rightarrow Ouest) et parfois même dans les huit directions si on considère pour chaque point son voisinage à huit.

Dans ce cas, on peut utiliser des différentiations, selon les directions choisies. Une direction est défini par l'angle de rotation par rapport à l'horizontale: [52]: $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$, π , $\frac{5\pi}{4}$, $\frac{3\pi}{2}$, $\frac{7\pi}{4}$.

2) Calcul du module du gradient:

Après avoir calculé les gradients de tous les points de l'image, on calcule les modules et les directions correspondants. On obtient alors une matrice Mod de même taille que l'image A, où Mod[i,j] représente le module du gradient au point A[i,j] de l'image. et une matrice Dir de même taille que A, où Dir[i,j] représente la direction du gradient au point A[i,j] de l'image.

- Dans le cas d'un gradient à deux directions [52]:

$$\text{Mod}[i,j] = \sqrt{\text{Grad}(i)^2 + \text{Grad}(j)^2} \quad (2.4)$$

$$\text{Dir}[i,j] = \text{Arctang}(\text{Grad}(j) / \text{Grad}(i)) \quad (2.5)$$

Afin de réduire le temps de calcul des modules de tous les gradients des pixels de l'image, on l'approche par:

$$\text{Mod}[i,j] = \text{Max}(|\text{Grad}(i)|, |\text{Grad}(j)|) \quad (2.6)$$

- Dans le cas d'un gradient à huit directions [39][52]:

$$\text{Mod}[i,j] = \text{Max}(|\text{Grad}_d(i,j)|) \quad d=1,\dots,8 \quad (2.7)$$

$$\text{Dir}[i,j] = 45 \times (d-1)^\circ \quad d=1,\dots,8 \quad (2.8)$$

3) Analyse du gradient de l'image et recherche des maxima locaux (seuillage):

On détermine sur la matrice Mod, les points où le gradient est maximal. Un maximum local du gradient dénote la présence d'un contour. On peut fixer a priori un seuil s, ainsi, un pixel A[i,j] est un contour si Mod[i,j]>s. En outre, on peut déterminer sur la matrice Dir, les directions de la plus grande pente du gradient, celle-ci est utilisée, dans les techniques de poursuite de contours [33].

1-1-2- Approche du Laplacien:

Cette approche est basée sur la dérivée seconde de la fonction de luminance, son passage à zéro dénote la présence d'un contour [40][33]. Elle se compose donc de trois étapes: le calcul du laplacien, la recherche des passages par zéro et le seuillage.

1) Calcul du Laplacien: Le Laplacien au niveau d'un pixel $A[i,j]$, se calcule par:

$$\text{Lap } [i,j] = \frac{\delta^2 G[i,j]}{\delta_i^2} + \frac{\delta^2 G[i,j]}{\delta_j^2} \quad (2.9)$$

$$\begin{aligned} \text{Lap}[i,j] &= (G[i-1, j]-G[i,j])-(G[i,j]-G[i+1,j]) + (G[i, j+1]-G[i,j])-(G[i,j]-G[i,j-1]) \\ &= b + d + f + h - 4e \end{aligned}$$

Fenêtre 3x3 de l'image	a	b	c	a = G[i-1,j-1];	b=G[i-1,j];	c=G[i-1,j+1]
	d	e	f	d = G[i,j-1];	e=G[i,j];	f=G[i,j+1]
	g	h	k	g = G[i+1,j-1];	h=G[i+1,j];	k=G[i+1,j+1]

Le laplacien calcule la différence entre le niveau de gris du pixel central et la moyenne des niveaux de gris des 4 pixels adjacents dans les directions horizontale et verticale. Ceci permet de renforcer les contours en atténuant les basses fréquences.

Le calcul du laplacien peut être réduit à un produit de convolution entre une fenêtre de l'image et un masque M de taille 3x3:

$$M = \begin{bmatrix} 0 & 1 & 0 \\ +1 & -4 & +1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Lap } [i,j] = V_{i,j} \ominus M \quad (2.10)$$

Où $V_{i,j}$: fenêtre (3x3) de l'image représentant le voisinage d'un pixel $A[i,j]$
et \ominus : symbole du produit de convolution

2) Passage par zéro: Les points contours se situent au points de passage par zéro de la fonction laplace.

3) Seuillage: La matrice Lap, obtenue précédemment contient des passages par zéro qui peuvent être aussi bien dûs à la présence des points contours qu'au bruit. Il est donc nécessaire de calculer le gradient de l'image et ainsi, tout passage par zéro ayant un faible gradient sera éliminé [40][33].

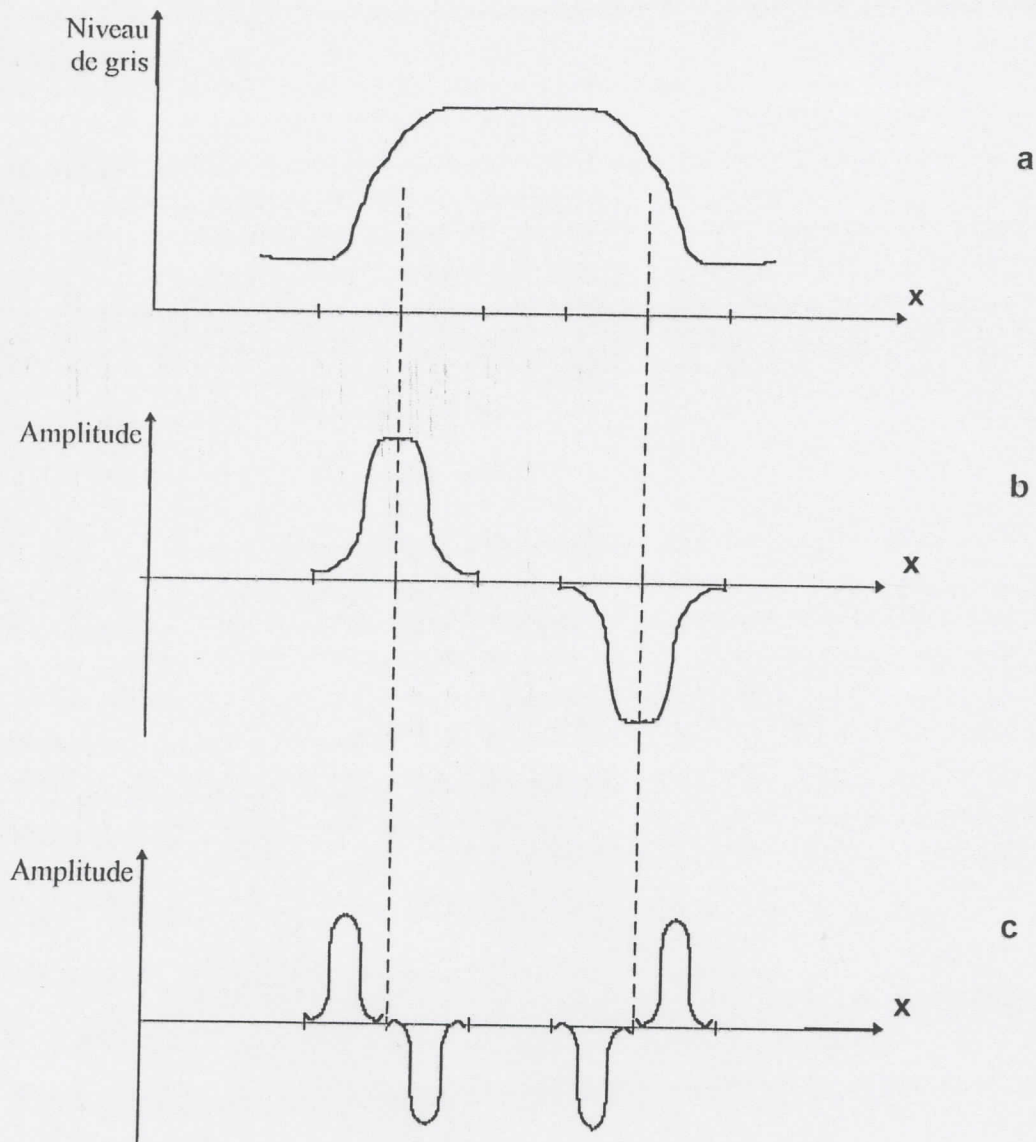


Figure 2.2 : Dérivation en présence d'un contour.
a: contour, b: gradient, c: laplacien

1-2- Détection de Contours par les Méthodes de Filtrage:

Le filtrage permet d'effectuer des transformations sur chaque pixel en tenant compte de son voisinage. Ces opérations appelées techniques de filtrage sont des **transformations locales** et sont classées en trois catégories:

- Méthodes de filtrage linéaire.
- Méthodes de filtrage non linéaire.
- Méthodes de filtrage adapté.

1-2-1- Méthodes de Filtrage Linéaire [33][39][46]:

L'application du filtre linéaire se fait en réalisant le produit de convolution du filtre avec une fenêtre de l'image:

1) **Filtre Passe bas:** Il élimine les hautes fréquences (bruit, irrégularité) et donne en sortie un signal de basse fréquence.

Un filtre passe bas associe à chaque pixel la moyenne des niveaux de gris de ces voisins, on peut utiliser plusieurs types de moyennes en pondérant différemment les niveaux de gris voisins. Ces coefficients doivent être positifs et leur somme doit être égale à 1. Le filtre passe bas présente l'inconvénient de traiter de la même manière, les signaux utiles et les bruits qui y sont superposés.

Exemple: Filtre passe Bas pour un masque de convolution 3x3

$$PB = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2) **Filtre passe haut:** Il possède les caractéristiques inverses du filtre passe bas. Il n'affecte pas les hautes fréquences mais atténue les basses fréquences (on l'utilise pour renforcer les contours).

- Ces coefficients peuvent être positifs ou négatifs.
- La somme de tous ces coefficients doit être égale à zéro.

Exemple :

$$PH = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3) Masque de Robert:

Il opère sur une fenêtre 2x2 pour chaque point d'où réduction du temps de calcul. Il utilise les dérivées diagonales pour estimer les gradients d'un point.

$G[i,j]=a$	$G[i,j+1]=b$
$G[i+1,j]=c$	$G[i+1,j+1]=d$

$$\delta_1 = G[i,j] - G[i+1,j+1] \quad \delta_2 = G[i+1,j] - G[i,j+1]$$

$$G'[i,j] = \sqrt{\delta_1^2 + \delta_2^2} = |\delta_1| + |\delta_2| = |a - d| + |c - b|$$

valeur estimée

Fenêtre 2x2

4) Masques de Prewitt:

Les masques de prewitt permettent l'extraction du gradient dans quatre directions principales. Chaque masque permet d'évaluer les transitions dans une direction particulière.

<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	0	-1	-1	-1	<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>0</td><td>-1</td><td>-1</td></tr> </table>	1	1	0	1	0	-1	0	-1	-1	<table border="1" style="border-collapse: collapse;"> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>-1</td><td>0</td></tr> </table>	0	1	1	-1	0	1	-1	-1	0
1	1	1																																					
0	0	0																																					
-1	-1	-1																																					
1	0	-1																																					
1	0	-1																																					
1	0	-1																																					
1	1	0																																					
1	0	-1																																					
0	-1	-1																																					
0	1	1																																					
-1	0	1																																					
-1	-1	0																																					

Horizontal

Vertical

Diagonal 45

Diagonal -45

5) Masques de Sobel :

L'application des opérateurs différentiels peut induire à une amplification du bruit dans l'image. Celui-ci peut être réduit en utilisant un opérateur de type intégrateur sur l'image.

$$G'[i,j] = V_{i,j} \ominus M_d \ominus M_{int} \tag{2.11}$$

Où $V_{i,j}$: fenêtre de l'image de centre le pixel (i,j).
 M_{int} : masque intégrateur.
 M_d : masque de dérivation selon une direction d.

Le produit de convolution étant associatif, on peut écrire:

$$G'[i,j] = V_{i,j} \ominus S_d \tag{2.12}$$

Où $S_d = M_d \ominus M_{int}$ appelé masque de Sobel suivant une direction d.

Masques de Sobel:

<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>-1</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> </table>	-1	-2	-1	0	0	0	1	2	1	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-2</td><td>0</td><td>2</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	0	1	-2	0	2	-1	0	1	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>-2</td><td>-1</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>2</td></tr> </table>	-2	-1	0	-1	0	1	0	1	2	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>0</td><td>-1</td><td>-2</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>2</td><td>1</td><td>0</td></tr> </table>	0	-1	-2	1	0	-1	2	1	0
-1	-2	-1																																					
0	0	0																																					
1	2	1																																					
-1	0	1																																					
-2	0	2																																					
-1	0	1																																					
-2	-1	0																																					
-1	0	1																																					
0	1	2																																					
0	-1	-2																																					
1	0	-1																																					
2	1	0																																					
Horizontal	Vertical	Diagonal 45	Diagonal -45																																				

1-2-2- Méthodes de Filtrage Non Linéaire[46]:

1) Filtre de la médiane: Pour une fenêtre contenant un nombre impair de pixels, on donne au pixel central la valeur médiane de l'ensemble de pixels de cette fenêtre.

fenêtre initiale		fenêtre finale																											
<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>32</td><td>35</td><td>33</td></tr> <tr><td>31</td><td>60</td><td>33</td></tr> <tr><td>30</td><td>29</td><td>34</td></tr> </table>	32	35	33	31	60	33	30	29	34	<table border="1" style="border-collapse: collapse; width: 100%; height: 15px;"> <tr> <td style="width: 10%;">29</td> <td style="width: 10%;">30</td> <td style="width: 10%;">31</td> <td style="width: 10%;">32</td> <td style="width: 10%; background-color: #cccccc;">33</td> <td style="width: 10%;">33</td> <td style="width: 10%;">34</td> <td style="width: 10%;">35</td> <td style="width: 10%;">60</td> </tr> </table>	29	30	31	32	33	33	34	35	60	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>32</td><td>35</td><td>33</td></tr> <tr><td>31</td><td>33</td><td>33</td></tr> <tr><td>30</td><td>29</td><td>34</td></tr> </table>	32	35	33	31	33	33	30	29	34
32	35	33																											
31	60	33																											
30	29	34																											
29	30	31	32	33	33	34	35	60																					
32	35	33																											
31	33	33																											
30	29	34																											

Application du filtre médian 3x3 sur une fenêtre 3x3

2) Filtres minimum et maximum: la valeur du pixel central est remplacée par la valeur minimale ou maximale de la fenêtre.

fenêtre initiale	fenêtre finale																		
<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>32</td><td>35</td><td>33</td></tr> <tr><td>31</td><td>60</td><td>33</td></tr> <tr><td>30</td><td>29</td><td>34</td></tr> </table>	32	35	33	31	60	33	30	29	34	<table border="1" style="border-collapse: collapse; width: 30px; height: 30px;"> <tr><td>32</td><td>35</td><td>33</td></tr> <tr><td>31</td><td>29</td><td>33</td></tr> <tr><td>30</td><td>29</td><td>34</td></tr> </table>	32	35	33	31	29	33	30	29	34
32	35	33																	
31	60	33																	
30	29	34																	
32	35	33																	
31	29	33																	
30	29	34																	

Application du filtre minimum 3x3 sur une fenêtre 3x3.

1-2-3- Méthodes de Filtrage Adapté [33]:

Le principe de ces méthodes est de trouver la plus grande ressemblance entre le voisinage de chaque point de l'image et un masque donné. On commence par convoluer l'image avec N masques (filtres) donnés, où chaque masque dépend d'une direction spécifique, puis sélectionner pour chaque point, le maximum des résultats obtenus par cette convolution, ceci permet de déterminer le filtre qui ressemble localement le plus à l'image. Parmi les opérateurs de filtrage adapté, on trouve:

1) Opérateur de Kirsh:

Cet opérateur associe un masque à chaque direction dans un voisinage 3x3. Chaque masque correspond à une direction entre le point central du voisinage (3x3) et un des huit voisins.

M1=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-5</td><td>-5</td><td>-5</td></tr> <tr><td>3</td><td>0</td><td>3</td></tr> <tr><td>3</td><td>3</td><td>3</td></tr> </table>	-5	-5	-5	3	0	3	3	3	3	M2=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>-5</td><td>-5</td></tr> <tr><td>3</td><td>0</td><td>-5</td></tr> <tr><td>3</td><td>3</td><td>3</td></tr> </table>	3	-5	-5	3	0	-5	3	3	3	M3=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>3</td><td>-5</td></tr> <tr><td>3</td><td>0</td><td>-5</td></tr> <tr><td>3</td><td>3</td><td>-5</td></tr> </table>	3	3	-5	3	0	-5	3	3	-5	M4=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>3</td><td>3</td></tr> <tr><td>3</td><td>0</td><td>-5</td></tr> <tr><td>3</td><td>-5</td><td>-5</td></tr> </table>	3	3	3	3	0	-5	3	-5	-5
-5	-5	-5																																									
3	0	3																																									
3	3	3																																									
3	-5	-5																																									
3	0	-5																																									
3	3	3																																									
3	3	-5																																									
3	0	-5																																									
3	3	-5																																									
3	3	3																																									
3	0	-5																																									
3	-5	-5																																									
M5=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>3</td><td>3</td></tr> <tr><td>3</td><td>0</td><td>3</td></tr> <tr><td>-5</td><td>-5</td><td>-5</td></tr> </table>	3	3	3	3	0	3	-5	-5	-5	M6=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>3</td><td>3</td><td>3</td></tr> <tr><td>-5</td><td>0</td><td>3</td></tr> <tr><td>-5</td><td>-5</td><td>3</td></tr> </table>	3	3	3	-5	0	3	-5	-5	3	M7=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-5</td><td>3</td><td>3</td></tr> <tr><td>-5</td><td>0</td><td>3</td></tr> <tr><td>-5</td><td>3</td><td>3</td></tr> </table>	-5	3	3	-5	0	3	-5	3	3	M8=	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-5</td><td>-5</td><td>3</td></tr> <tr><td>-5</td><td>0</td><td>3</td></tr> <tr><td>3</td><td>3</td><td>3</td></tr> </table>	-5	-5	3	-5	0	3	3	3	3
3	3	3																																									
3	0	3																																									
-5	-5	-5																																									
3	3	3																																									
-5	0	3																																									
-5	-5	3																																									
-5	3	3																																									
-5	0	3																																									
-5	3	3																																									
-5	-5	3																																									
-5	0	3																																									
3	3	3																																									

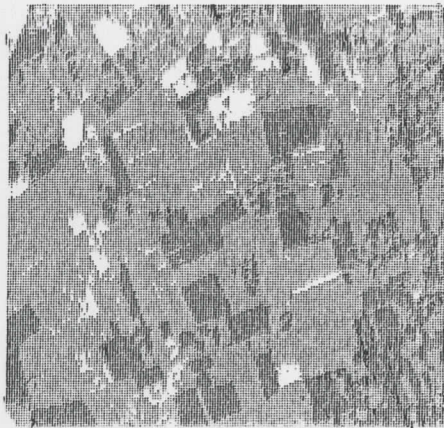
2) Opérateur moyenne isotropique: Ces masques sont déduits des masques de Sobel, par une pondération permettant de compenser l'effet de distribution des points diagonaux par rapport aux points centraux:

<u>Masques:</u>	<table style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>$-\sqrt{2}$</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>$\sqrt{2}$</td><td>1</td></tr> </table>	-1	$-\sqrt{2}$	-1	0	0	0	1	$\sqrt{2}$	1	<table style="border: 1px solid black; padding: 5px;"> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>$-\sqrt{2}$</td><td>0</td><td>$\sqrt{2}$</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	0	1	$-\sqrt{2}$	0	$\sqrt{2}$	-1	0	1	<table style="border: 1px solid black; padding: 5px;"> <tr><td>$-\sqrt{2}$</td><td>-1</td><td>0</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>$\sqrt{2}$</td></tr> </table>	$-\sqrt{2}$	-1	0	-1	0	1	0	1	$\sqrt{2}$	<table style="border: 1px solid black; padding: 5px;"> <tr><td>0</td><td>-1</td><td>$-\sqrt{2}$</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>$\sqrt{2}$</td><td>1</td><td>0</td></tr> </table>	0	-1	$-\sqrt{2}$	1	0	-1	$\sqrt{2}$	1	0
-1	$-\sqrt{2}$	-1																																						
0	0	0																																						
1	$\sqrt{2}$	1																																						
-1	0	1																																						
$-\sqrt{2}$	0	$\sqrt{2}$																																						
-1	0	1																																						
$-\sqrt{2}$	-1	0																																						
-1	0	1																																						
0	1	$\sqrt{2}$																																						
0	-1	$-\sqrt{2}$																																						
1	0	-1																																						
$\sqrt{2}$	1	0																																						
	Horizontal	Vertical	Diagonal 45	Diagonal -45																																				

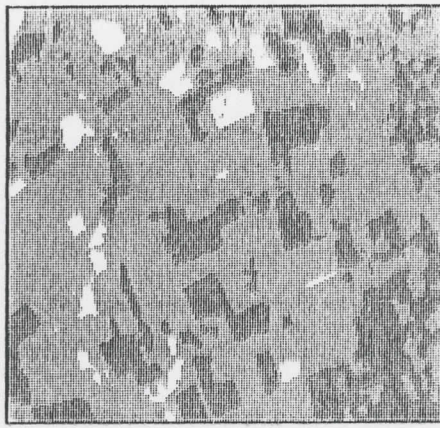
1-3- Application des Filtres:

Données: L'image utilisée pour l'application des différents filtres, est une image SPOT XS1, de résolution spatiale 20x20m. Elle représente un paysage hétérogène de la région de l'Atlas Blidéen, variant du relief montagneux aux zones urbaines et à la couverture agricole. Ces filtres sont appliqués sur une fenêtre de l'image de taille 170x170 pixels.

1-3-1- Affichage des Images:



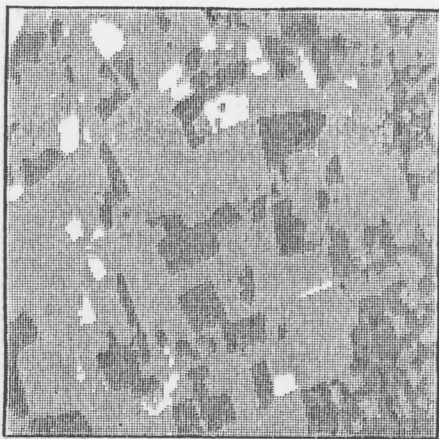
1- Image originale



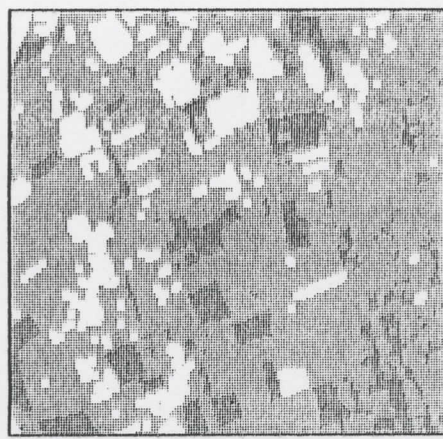
2- Application du filtre passe-bas



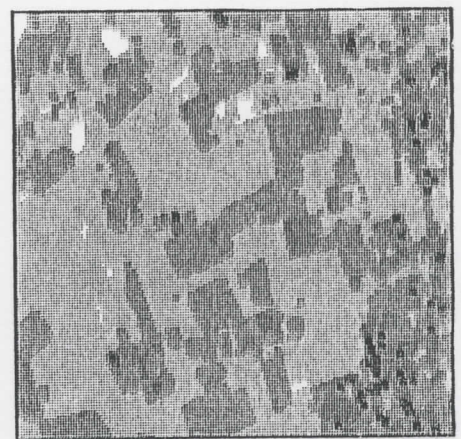
3- Application du filtre passe-haut



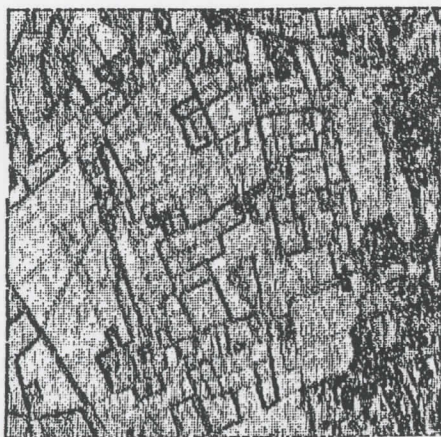
4- Application du filtre médian



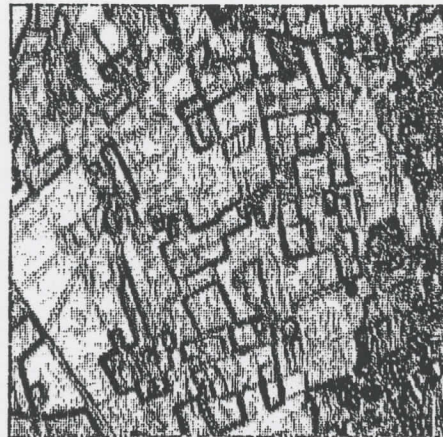
5- Application du filtre maximum



6- Application du filtre minimum



7- Application de l'opérateur de ROBERT



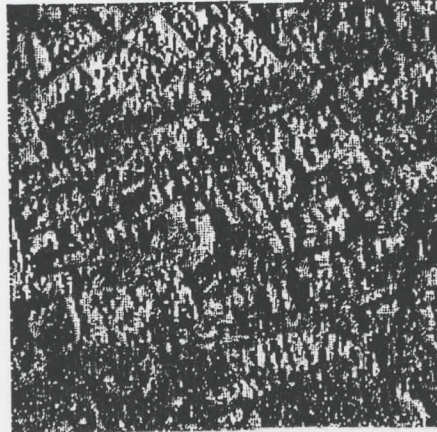
8- Application de l'opérateur de PREWITT



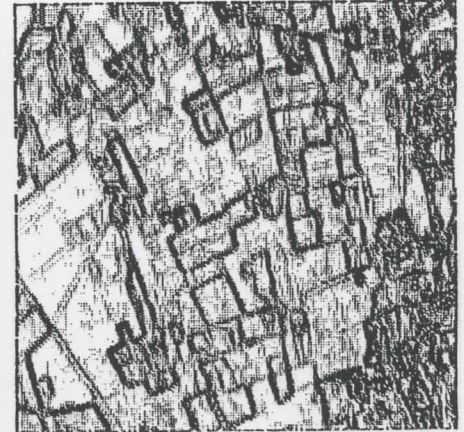
9- Application de l'opérateur de SOBEL



10- Application du Laplacien



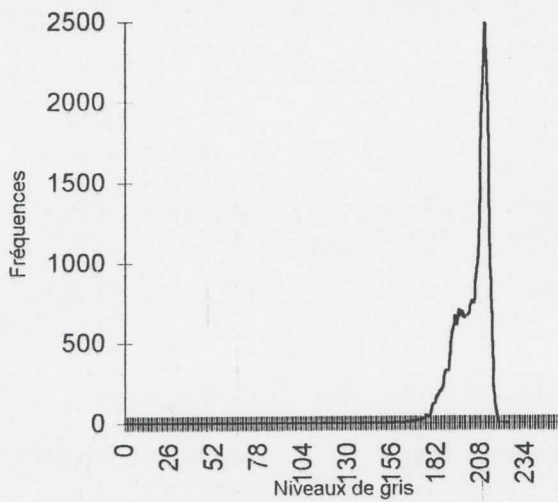
11- Application de l'opérateur de KIRSH



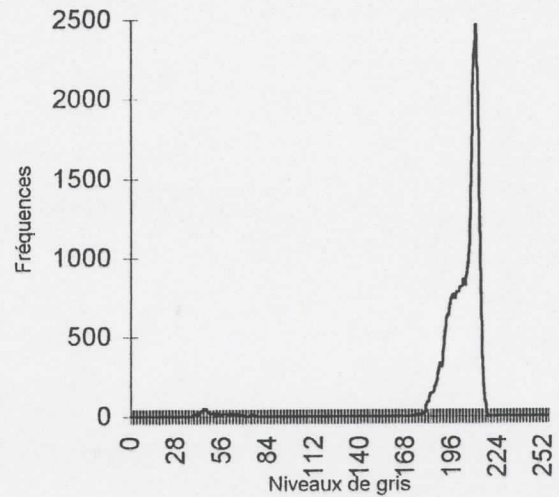
12- Application de l'opérateur Isotropique

Figure 2.3 : Application des filtres sur une image satellitaire.
(1: Image originale, 11-12: Images filtrées)

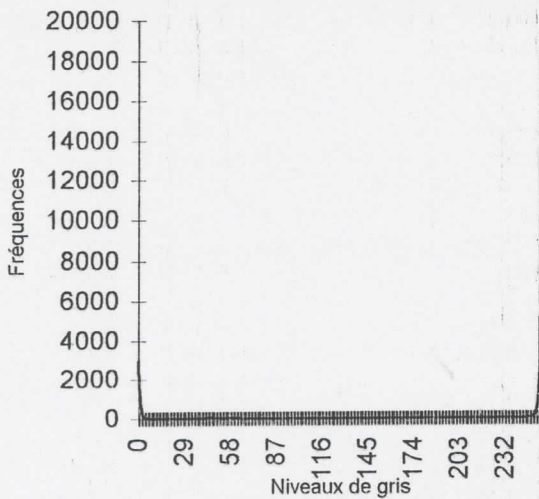
1-3-2- Affichage des Histogrammes des niveaux de gris:



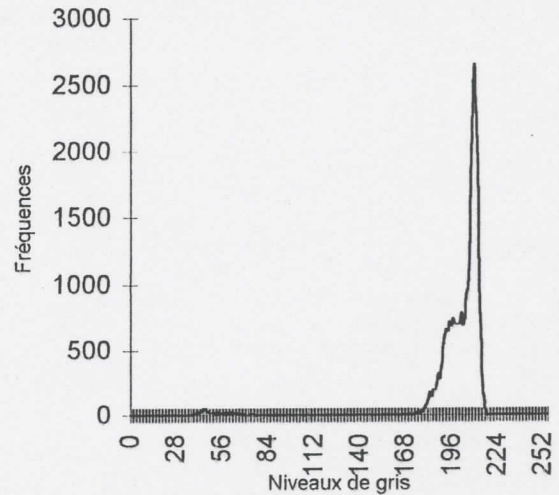
1- Histogramme de l'image originale



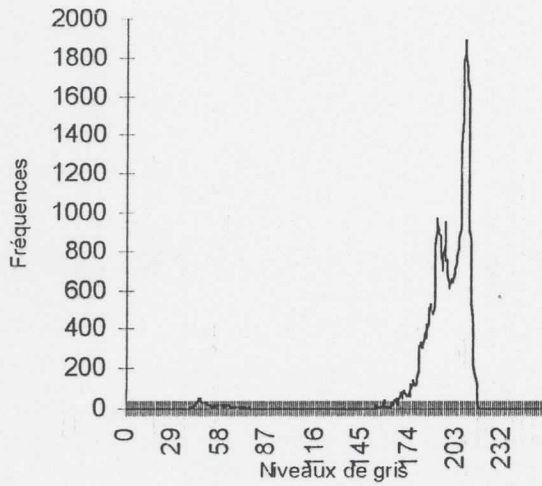
2- Histogramme de l'image traitée par le filtre passe bas



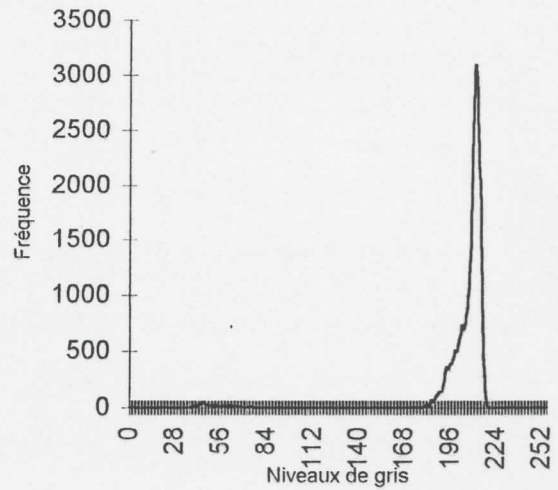
3- Histogramme de l'image traitée par le filtre passe haut



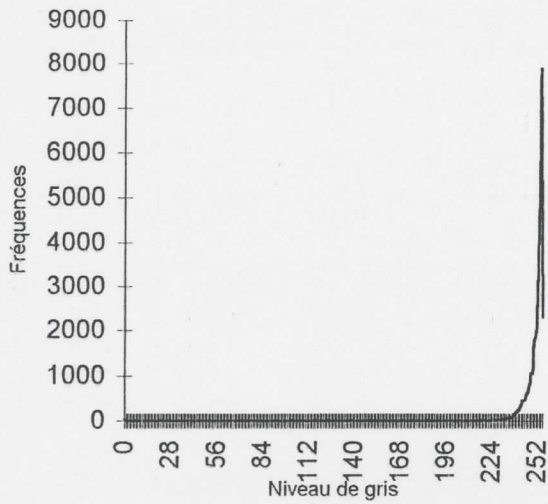
4- Histogramme de l'image traitée par le filtre médian



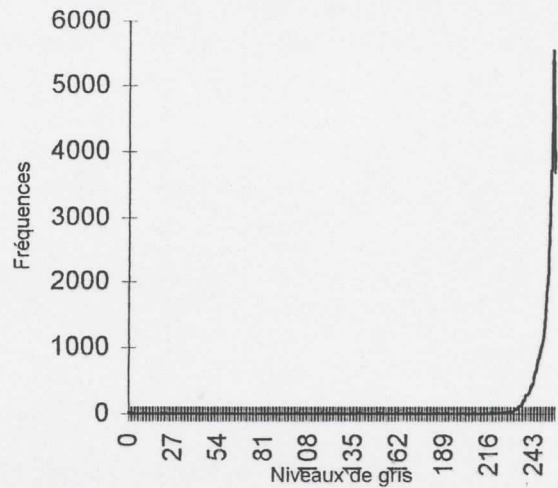
5- Histogramme de l'image traitée par le filtre maximum



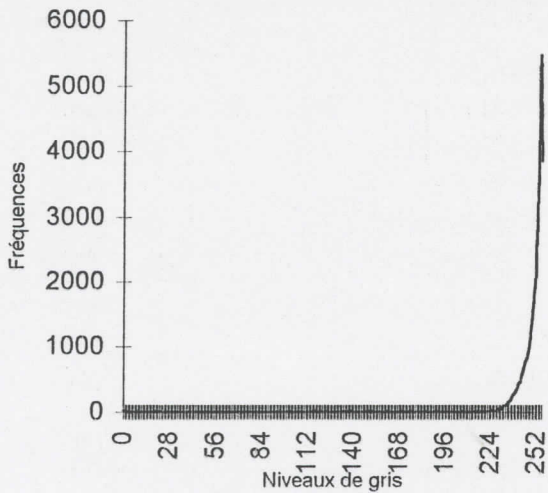
6- Histogramme de l'image traitée par le filtre minimum



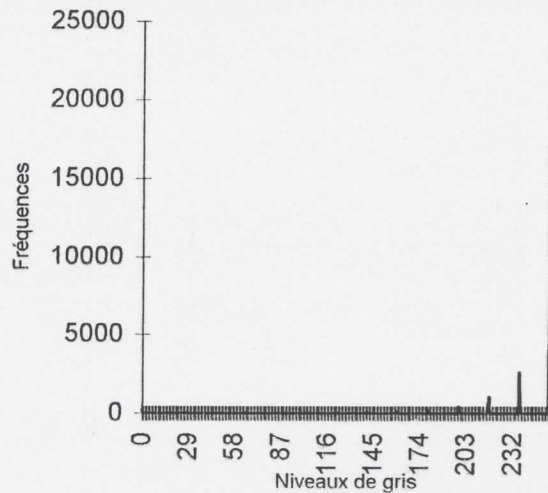
7- Histogramme de l'image traitée par l'opérateur de ROBERT



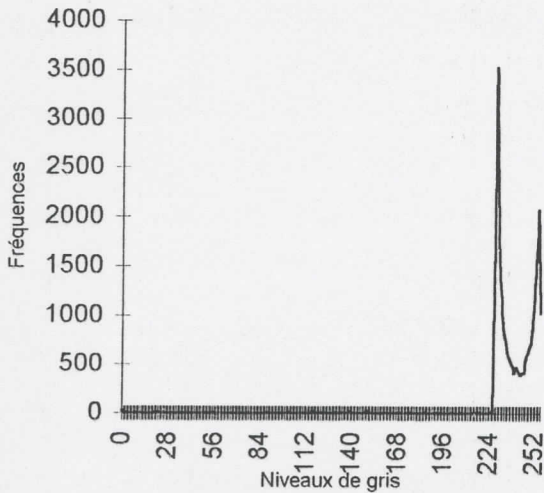
8- Histogramme de l'image traitée par l'opérateur de PREWITT



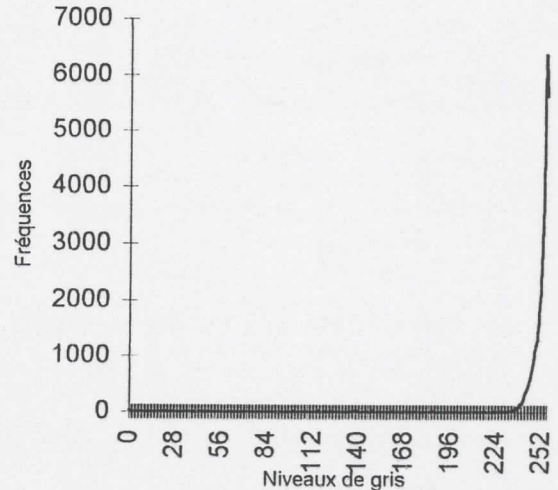
9- Histogramme de l'image traitée par l'opérateur de SOBEL



10- Histogramme de l'image traitée par le Laplacien



11- Histogramme de l'image traitée par l'opérateur de KIRSH



12- Histogramme de l'image traitée par l'opérateur isotropique

Figure 2.4 : Histogrammes des images traitées par les filtres.
(1: Histogramme de l'image originale, 11-12: Histogrammes des images filtrées).

1-3-3- Analyse Comparative: Les différents filtres présentés, sont comparés selon la visualisation des images résultantes de ces filtres, leurs histogrammes correspondants ainsi que l'espace mémoire et le temps d'exécution nécessaires au déroulement de chaque méthode.

- Analyse visuelle des images:

<i>Masques</i>	<i>Analyse visuelle</i>
1) Filtre Passe-bas	- Disparition des points isolés dans l'image.
2) Filtre Passe-haut	- Augmentation du contraste. - Apparition de contours amincis dans l'image.
3) Filtre médian	- Disparition des pixels isolés. - Légère augmentation du contraste.
4) Filtre maximum	- Disparition des pixels isolés. - Augmentation du nombre de tâches blanches.
5) Filtre minimum	- Disparition des pixels isolés. - Augmentation du nombre de tâches noires.
6) Robert	- Apparition de contours dans l'image.
7) Prewitt	- Apparition de contours forts dans l'image.
8) Sobel	- Apparition de contours forts dans l'image.
9) Laplacien	- Augmentation du contraste. - Apparition de contours amincis et discontinus.
10) Kirsh	- Diminution du contraste. - Image floue.
11) Moyenne isotropique	- Apparition de contours forts dans l'image.

Tableau 2.1: Application des filtres- Analyse visuelle.

- **Analyse des histogrammes :**

<i>Masques</i>	<i>Analyse des histogrammes</i>
1) Filtre Passe-bas	- Histogramme ne présentant pas une différence avec l'histogramme original
2) Filtre Passe-haut	- Histogramme présentant deux pics amincis, correspondant à deux classes de niveaux de gris respectifs noir et blanc.
3) Filtre médian	- Histogramme ne présentant pas une différence avec l'histogramme original
4) Filtre maximum	- Histogramme semibimodal. Les modes sont dans l'intervalle [200,220]
5) Filtre minimum	- Histogramme aminci avec léger élargissement pour la classe [180-200]
6) Robert	- Histogramme en pic vers la droite représentant une émergence d'une classe par rapport à d'autres.
7) Prewitt	- Histogramme en pic vers la droite représentant une émergence d'une classe par rapport à d'autres.
8) Sobel	- Histogramme en pic vers la droite représentant une émergence d'une classe par rapport à d'autres.
9) Laplacien	- Histogramme à plusieurs pics.
10) Kirsh	- Histogramme bimodal, de modes 230 et 245.
11) Moyenne isotropique	- Histogramme en pic vers la droite représentant une émergence d'une classe par rapport à d'autres.

Tableau 2.2: Application des filtres- Analyse des histogrammes.

- **Temps d'exécution et espace mémoire:**

<i>Masques</i>	<i>Temps d'exécution (ms)</i>	<i>Espace mémoire (octets)</i>
1) Filtre Passe-bas	500	30400
2) Filtre Passe-haut	490	30400
3) Filtre médian	650	30500
4) Filtre maximum	270	30470
5) Filtre minimum	270	30470
6) Robert	700	30800
7) Prewitt	220	30840
8) Sobel	160	30850
9) Laplacien	440	30900
10) Kirsh	2530	30870
11) Moyenne isotropique	6920	30860

Tableau 2.3: Application des filtres- Temps d'exécution et espace mémoire.

1-3-4- Conclusion:

En observant les images obtenues par les filtres passe-bas, maximum, minimum et médian, nous constatons que ces filtres permettent un lissage de l'image, ils sont utiles dans l'étape prétraitement plutôt que dans l'étape de segmentation. Cependant ils présentent un inconvénient qui est le risque de perdre des données utiles qu'ils auraient traitées comme étant un bruit. Cette perte d'informations est nettement visible sur les images obtenues par le filtre maximum et minimum.

En comparant les temps d'exécution de chacun de ces quatre filtres, on constate que les filtres maximum et minimum présentent une rapidité dans l'exécution relativement au filtre passe-bas et au filtre médian. Cependant, l'écart entre les différents temps d'exécution n'étant pas important à comparer avec la qualité des résultats, le filtre passe-bas est retenu comme étant le meilleur pour l'opération de prétraitement.

L'application des masques de Robert, Prewitt, Sobel, et moyenne isotropique, présente de bons résultats quant à la segmentation de l'image. En effet, des contours apparaissent sur l'image, décrivant les différentes régions homogènes. L'apparition des pics sur les histogrammes correspondant à ces masques, confirme cette constatation. Ces contours sont plus nets sur les images résultant des masques de Prewitt, Sobel, et moyenne isotropique, que sur l'image obtenue par le masque de Robert. Cela est dû au fait que celui-ci opère seulement dans les deux directions diagonales.

Les contours obtenus par ces différentes applications sont d'une épaisseur non négligeable et nécessitent une opération d'amincissement. Dans le (§ 4.1.3 p. 53), nous présentons une méthode d'amincissement de contours (squelettisation).

Une comparaison des temps d'exécution de ces différents masques, montre que l'opérateur isotropique présente le temps d'exécution le plus long. Ceci peut être un paramètre exclusif pour l'opérateur isotropique surtout dans le cas où nous avons à traiter des images de plus grande taille.

Les filtres passe-haut et le laplacien présentent une très bonne extraction de contours dans l'image. Les contours sont parfaitement amincis, mais présentent des discontinuités ne correspondant pas à la réalité. Ces discontinuités sont dues à la sensibilité de ces deux filtres au bruit.

L'opérateur de Kirsh, donne un résultat non satisfaisant, l'image obtenue est peu contrastée et floue, correspondant peu à l'image originale, il ne peut être retenu pour la segmentation des images.

Pour les différents masques, le paramètre espace mémoire, n'est pas sélectif. Car toutes ces procédures sont sous formes d'un produit de convolution entre un masque et une fenêtre de l'image de même taille, sauf pour les filtres minimum, maximum et médian, qui sont basés sur une opération de comparaison entre les pixels d'un même voisinage.

2- Segmentation par Seuillage Automatique:

La segmentation par seuillage consiste à comparer un certain attribut scalaire ou vectoriel calculé en chaque point de l'image avec un ou plusieurs seuils scalaires ou vectoriels, en vue d'attribuer le point considéré à une classe. Elle s'adresse à des problèmes simples tels la classification des points d'une image non texturée en «fond» et «objet» [52].

La segmentation par seuillage des images à niveau de gris, utilise l'information radiométrique (niveau de gris) de l'image sans tenir compte de l'aspect contextuel. Elle utilise aussi, les informations contenues dans l'histogramme de l'image.

Dans le cas d'un histogramme bimodal, on effectue un seuillage binaire, en sélectionnant le seuil (niveau de gris) qui sépare les deux modes de l'histogramme et l'image sera partagée en deux classes (Figure 2.5).

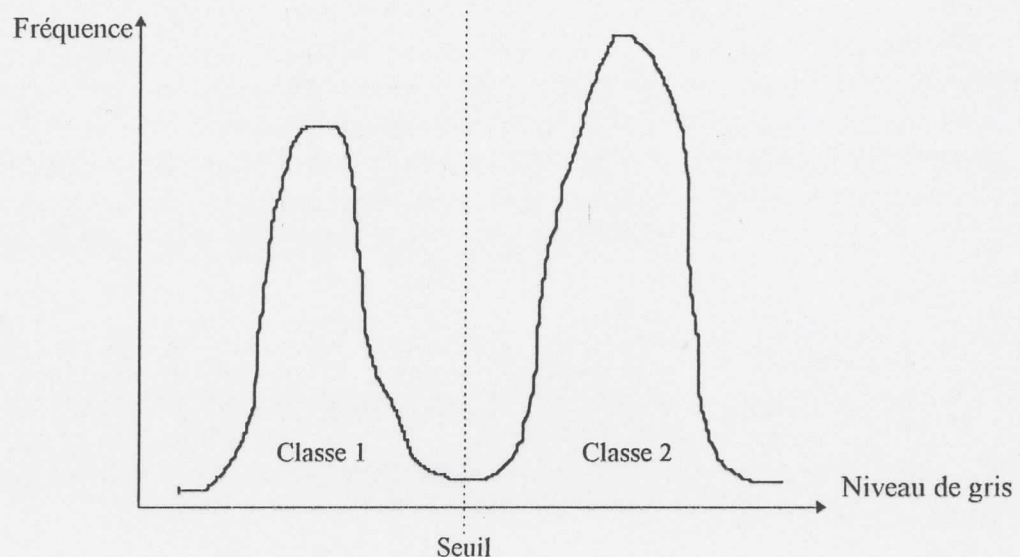


Figure 2.5 : Histogramme bimodal.

En présence d'un histogramme multimodal, on conclut que nous avons à détecter un ou plusieurs objets sur un ou plusieurs fonds avec des distributions des niveaux de gris des points de surface des objets et du fond ayant une faible variance et des moyennes différentes (Figure 2.6). Dans ce cas, on utilise un multiseuillage qui consiste à sélectionner les seuils séparant deux modes consécutifs et partager l'image en plusieurs classes en fonction de ces seuils.

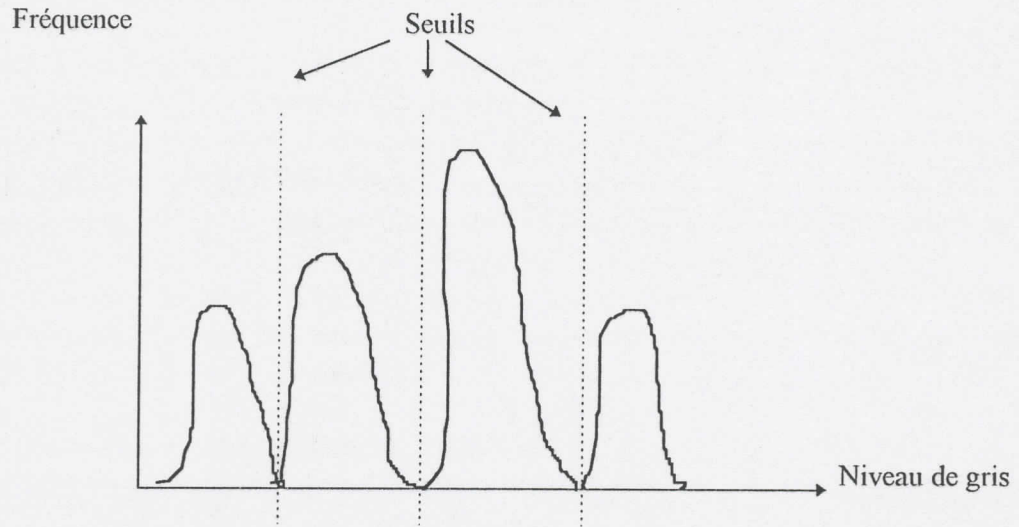


Figure 2.6 : Histogramme multimodal.

Généralement les images satellitaires présentent un histogramme unimodal. En réalité ces images sont composées de plusieurs objets, dont les histogrammes des niveaux de gris respectifs se recouvrent (Figure 2.7), pour donner un histogramme global unimodal [19]. Ce phénomène de recouvrement rend plus compliqué la détermination des seuils et par conséquent l'application du seuillage sur ces images devient plus difficile.



Figure 2.7 : Contribution des histogrammes de niveaux de gris de plusieurs objets à l'historgramme global de l'image.

• **Définition formelle du seuillage:**

- Soit $A(L,K)$ une image numérique à L lignes et K colonnes,
 - soit $G(L,K)$ la matrice représentant les niveaux de gris de l'image
 La segmentation de l'image, en M classes C_m $m=1, \dots, M$ par la méthode du seuillage, revient à déterminer $M-1$ seuils tels que [33]:

$$\begin{aligned}
 A[i,j]_{i=1, \dots, L; j=1, \dots, K} &\in C_1 && \text{si } G[i,j] \leq S_1 \\
 A[i,j]_{i=1, \dots, L; j=1, \dots, K} &\in C_m \quad m=2, \dots, M-1 && \text{si } S_{m-1} < G[i,j] \leq S_m \\
 A[i,j]_{i=1, \dots, L; j=1, \dots, K} &\in C_M && \text{si } G[i,j] > S_{M-1}
 \end{aligned}$$

2-1- Méthodes de Calcul des Seuils:

On désigne par X la variable aléatoire dénotant le niveau de gris des pixels de l'image. A partir de l'histogramme H des niveaux de gris de l'image A , on définit la fonction de densité des probabilités des niveaux de gris de l'image A qu'on note $f(x)$.

$$H[x] = \text{Card} \{A[i,j]_{i=1, \dots, L; j=1, \dots, K} / G[i,j]=x\} \quad \forall x \in N \cap [0,255] \quad (2.13)$$

$$f(x) = \frac{H[x]}{LxK} = \frac{\text{Card}\{A[i,j]_{i=1, \dots, L; j=1, \dots, K} / G[i,j]=x\}}{LxK} \quad \forall x \in N \cap [0,255] \quad (2.14)$$

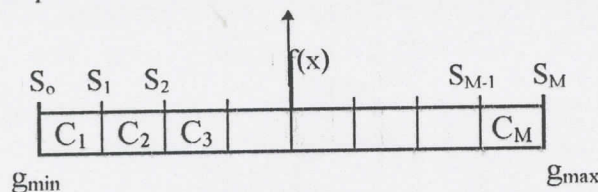
Où $\text{Card} \{E\}$ représente la cardinalité de E et N représente l'ensemble des entiers naturels.

Dans les deux méthodes de seuillage, que nous présentons, la fonction $f(x)$ permet de déterminer les paramètres statistiques de l'image, tels que la moyenne et la variance des niveaux de gris.

2-1-1- Méthode de coalescence ISODATA [33]:

2-1-1-1- Principe de la Méthode:

L'algorithme ISODATA, permet de calculer les différents seuils indépendamment de la répartition géométrique des niveaux de gris dans l'image. On définit tout d'abord $[g_{\min}, g_{\max}]$ le plus petit intervalle contenant les valeurs non nulles de $f(x)$, qu'on partage en M classes.



Pour chaque classe $C_m, m=1, \dots, M$, on calcule sa moyenne arithmétique moy_m , en supposant que la densité de probabilité $f(x)$ est uniforme entre g_{min} et g_{max} . Les différents seuils sont ensuite calculés en fonction de ces moyennes.

$$\left. \begin{aligned} S_0 &= g_{min} \\ S_m &= \left[\frac{moy_m + moy_{m+1}}{2} \right] \quad m=1, \dots, M-1 \\ S_M &= g_{max} \end{aligned} \right\} \quad (2.15)$$

Où $[a]$ représente la partie entière du nombre a .

- Les points $A[i,j]$ de l'image sont classés selon la formule:

$$S_{m-1} < G[i,j] \leq S_m \Rightarrow A[i,j] \quad i=1, \dots, L; j=1, \dots, K \in C_m \quad m=1, \dots, M$$

Dans l'étape suivante, on met à jour les moyennes en fonction des résultats de la classification, par:

$$moy_m = \left[\frac{\sum_{x \in I_m} x f(x)}{\sum_{x \in I_m} f(x)} \right] ; \quad I_m = N \cap [S_{m-1}, S_m] \quad (2.16)$$

Si aucune moyenne n'a changé de valeurs alors arrêter sinon le processus du calcul des seuils est réitéré en fonction des nouvelles moyennes.

2-1-1-2- Algorithme:

Etape1: Déterminer l'intervalle $[g_{min}, g_{max}]$;
 Partager $[g_{min}, g_{max}]$ en M classes équidistantes;
 Calculer les moyennes arithmétiques $moy_m \quad m=1, \dots, M$ de chaque classe;

Etape 2: (*Evaluation des seuils et classification*)

$S_0 = g_{min} ; \quad S_m = \left[\frac{moy_m + moy_{m+1}}{2} \right] \quad m=1, \dots, M-1 ; \quad S_M = g_{max} ;$

Pour chaque pixel $A[i,j]$ de l'image

Faire Si $S_{m-1} < G[i,j] \leq S_m$
 Alors $A[i,j] \in C_m \quad m=1, \dots, M$
 Fsi;

Fait;

Etape 3: Mise à jour des moyennes moy_m

$moy_m = \left[\frac{\sum_{x \in I_m} x f(x)}{\sum_{x \in I_m} f(x)} \right] ; \quad I_m = N \cap [S_{m-1}, S_m]$

Si toute moyenne moy_m reste invariante
 Alors fin
 Sinon revenir à l'étape 2

Fsi;

Fin;

Algorithme 2.1: Isodata.

2-1-2- Calcul des seuils par l'analyse discriminante [33]:

2-1-2-1- Principe de l'analyse discriminante:

Cette méthode repose sur le principe de séparation des classes C_m $m=1, \dots, M$. D'une part, on éloigne le plus possible, les classes les unes des autres, ce qui revient à maximiser la variance interclasse et d'autres part on regroupe au mieux chaque classe autour de sa moyenne c'est à dire minimiser la variance intraclasse. Le meilleur seuillage correspondrait donc, à une minimisation de la variance intraclasse ou la maximisation de la variance interclasse.

Dans ce paragraphe, on présente la méthode de segmentation de l'image originale en M classes disjointes en maximisant un critère statistique de séparabilité entre classes.

1- On détermine $[g_{\min}, g_{\max}]$ le plus petit intervalle contenant les valeurs non nulles de $f(x)$ et on prend $S_0 = g_{\min}$ et $S_M = g_{\max}$

2- On calcule la valeur moyenne μ et la variance σ^2 de la variable aléatoire X

$$\mu = \sum_{x \in N \cap [0,255]} x f(x) \quad (2.17)$$

$$\sigma^2 = \sum_{x \in N \cap [0,255]} (x - \mu)^2 f(x) \quad (2.18)$$

3- La probabilité qu'un point de l'image appartienne à la classe C_m est:

$$w_m = \sum_{x=S_{m-1}}^{S_m} f(x) \quad m=1, \dots, M \quad (2.19)$$

4- Pour chaque classe C_m , on a la valeur moyenne des niveaux de gris:

$$\mu(S_m) = \sum_{x=S_{m-1}}^{S_m} x f(x) \quad (2.20)$$

5- Ces valeurs sont normalisées par la probabilité d'appartenance à chaque classe:

$$\mu_m = \frac{\mu(S_m)}{w_m} \quad (2.21)$$

6- On déduit la variance σ_m^2 séparant deux classes C_m et C_{m+1} :

$$\begin{aligned} \sigma_m^2 &= w_m(\mu_m - \mu)^2 + w_{m+1}(\mu_{m+1} - \mu)^2 \\ &= w_m \left(\frac{\mu(S_m)}{w_m} - \mu \right)^2 + w_{m+1} \left(\frac{\mu(S_{m+1})}{w_{m+1}} - \mu \right)^2 \\ &= \frac{(\mu(S_m) - \mu w_m)^2}{w_m} + \frac{(\mu(S_{m+1}) - \mu w_{m+1})^2}{w_{m+1}} \end{aligned}$$

7- Pour chaque couple de classes C_m, C_{m+1} $m=1, \dots, M-1$, trouver le seuil S_m^* maximisant la variance σ_m^2

Application à une segmentation en deux classes:

Il s'agit d'effectuer un seuillage en deux classes C_1 et C_2 .

2-1-2-2- Algorithme:

$$\mu \leftarrow \sum_{x \in N \cap [0,255]} xf(x) \qquad \sigma^2 \leftarrow \sum_{x \in N \cap [0,255]} (x-\mu)^2 f(x)$$

$$S_0 \leftarrow g_{\min} ; \quad S_2 \leftarrow g_{\max}$$

Pour toute valeur de $S_1 \in [S_0, S_2]$ faire:

La probabilité qu'un point appartienne à C_1 respectivement à C_2 est :

$$w_1 = \sum_{x=S_0}^{S_1} f(x) \qquad w_2 = \sum_{x=S_1+1}^{S_2} f(x)$$

La valeur moyenne des niveaux de gris de C_1 respectivement de C_2 est :

$$\mu(S_1) = \sum_{x=S_0}^{S_1} xf(x) \qquad \mu(S_2) = \sum_{x=S_1+1}^{S_2} xf(x) = \mu - \mu(S_1)$$

Normalisation des moyennes:

$$\mu_1 = \frac{\mu(S_1)}{w_1} \qquad \mu_2 = \frac{\mu(S_2)}{w_2} = \frac{\mu - \mu(S_1)}{1-w_1}$$

La variance σ^2_1 séparant les deux classes C_1 et C_2 est :

$$\begin{aligned} \sigma^2_1 &= w_1(\mu_1 - \mu)^2 + w_2(\mu_2 - \mu)^2 \\ &= w_1 \left(\frac{\mu(S_1)}{w_1} - \mu \right)^2 + (1-w_1) \left(\frac{\mu - \mu(S_1)}{1-w_1} - \mu \right)^2 \\ &= \frac{(\mu(S_1) - \mu w_1)^2}{w_1(1-w_1)} \end{aligned}$$

Retenir $S_1^* \in [S_0, S_2]$ telle que σ^2_1 est maximale.

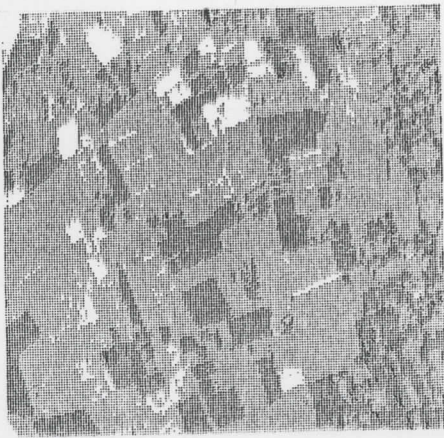
Fin .

Algorithme 2.2: Analyse discriminante. Cas de deux classes.

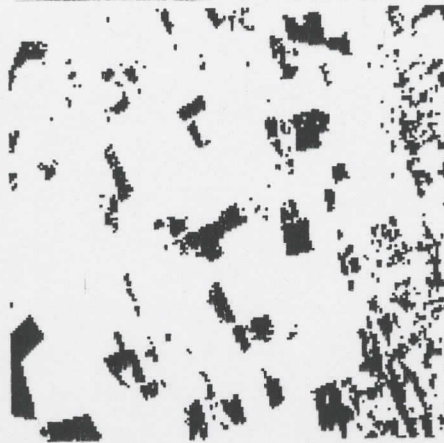
2-2- Application des méthodes de seuillage:

Données: L'algorithme Isodata et l'algorithme de l'analyse discriminante dépendent des paramètres statistiques de l'image qui sont la variance et la moyenne des niveaux de gris des pixels de l'image. Pour cela, ces deux algorithmes sont appliqués sur des fenêtres de différentes tailles, de l'image SPOT XS1, représentant la région de l'Atlas Blidéen de résolution spatiale 20x20m.

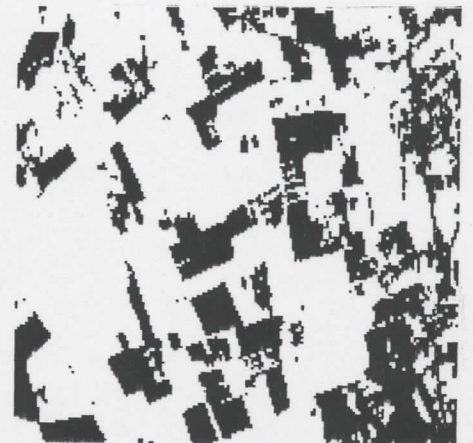
2-2-1 - Affichage des Images et des Histogrammes:



1- Image originale 170x170



2- Traitement de (1) par Isodata



3- Traitement de (1) par l'analyse discriminante



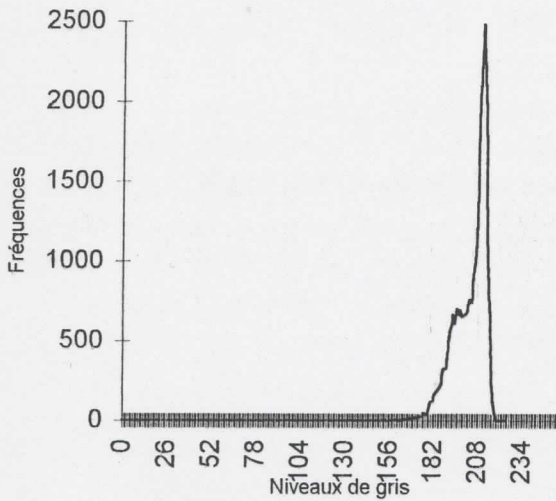
4- Fenêtre 50x50 de l'image originale



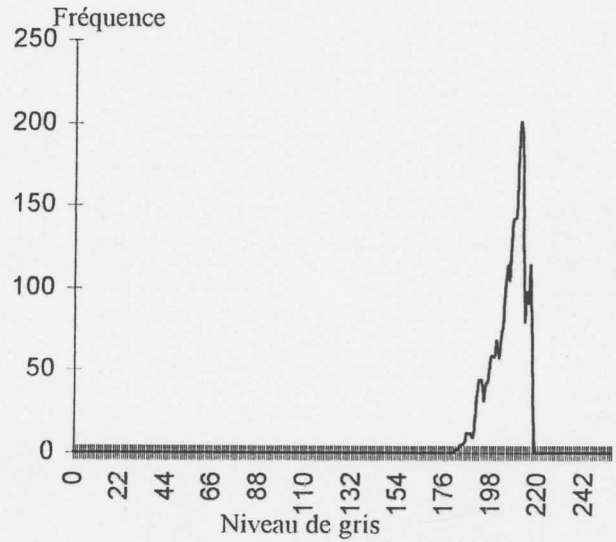
5- Traitement de (4) par Isodata



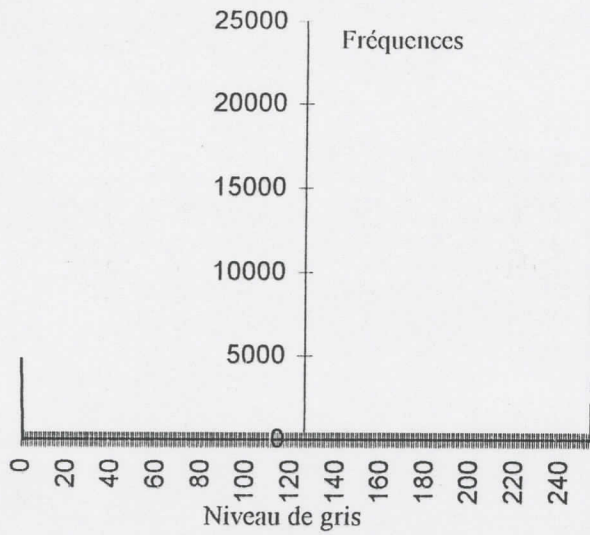
6- Traitement de (4) par l'analyse discriminante



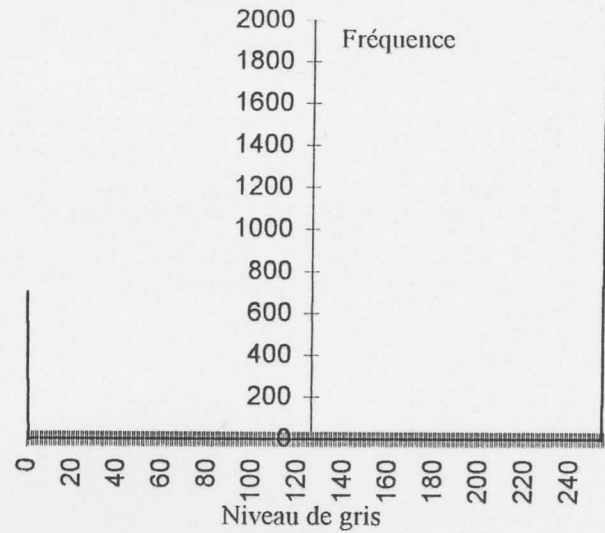
7-) Histogramme de l'image (1)



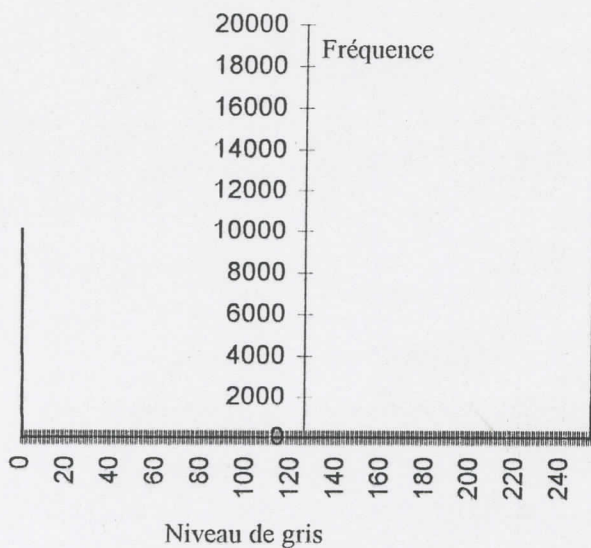
10-) Histogramme de l'image (4)



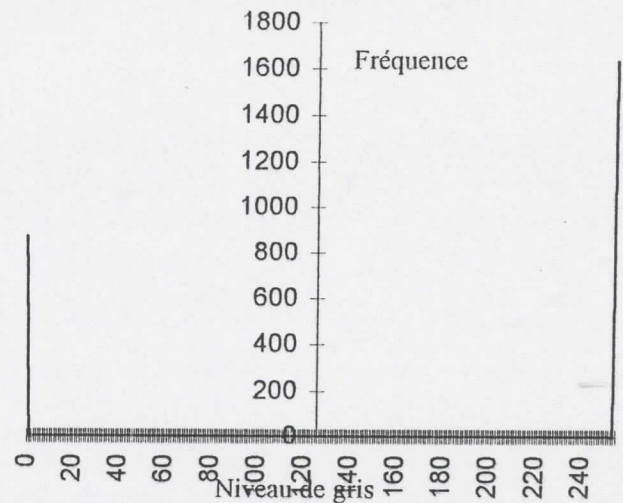
8-) Histogramme de l'image (2)



11-) Histogramme de (5)



9-) Histogramme de l'image (3)



12-) Histogramme de (6)

Figure 2.8 : Application du seuillage.
(1-6): Images - (7-12): Histogrammes des niveaux de gris.

2-2-2- Analyse comparative des résultats:

Cette analyse est basée sur l'appréciation visuelle des images, les histogrammes des niveaux de gris des images, les paramètres statistiques de l'image ainsi que sur l'espace mémoire et le temps d'exécution nécessaires aux deux méthodes en fonction de la taille de l'image à traiter.

- Analyse visuelle des images:

Cas	Méthodes	Taille de l'image	Analyse visuelle des images
1-)	Isodata	170x170	- Apparition sur l'image de deux classes distinctes correspondants aux régions de l'image.
2-)	Isodata	50x50	- L'image est segmentée en deux classes. Amélioration du résultat de classification par rapport au cas précédent, par l'apparition de nouvelles zones distinctes de l'image.
3-)	Analyse discriminante	170x170	- L'image est segmentée en deux classes.
4-)	Analyse discriminante	50x50	Amélioration du résultat par rapport au cas précédent, par l'apparition de zones qui n'étaient pas mises en évidence dans le cas 3.

Tableau 2.4: Application des méthodes de seuillage. Analyse visuelle des images.

- Paramètres statistiques de l'image :

Cas	Méthodes	Taille de l'image	Paramètres statistiques de l'image originale	Paramètres statistiques de l'image résultat	Paramètres statistiques de C_1	Paramètres statistiques de C_2
1-)	Isodata	170x170	moyenne= 51	$S_1 = 102$	moy ₁ = 48.44 var ₁ = 32.88	moy ₂ = 67.53 var ₂ = 31.04
2-)	Isodata	50x50	moyenne= 49	$S_1 = 74$	moy ₁ = 45.17 var ₁ = 13.94	moy ₂ = 58.99 var ₂ = 21.73
3-)	Analyse discriminante	170x170	$\mu = 51$ $\sigma^2 = 82$	$S_1^* = 54$ $\sigma^{2*} = 60.46$	$\mu_1 = 45.89$ var ₁ = 11.23	$\mu_2 = 62.20$ var ₂ = 30.57
4-)	Analyse discriminante	50x50	$\mu = 49$ $\sigma^2 = 54$	$S_1^* = 51$ $\sigma^{2*} = 38.43$	$\mu_1 = 44.99$ var ₁ = 10.72	$\mu_2 = 57.46$ var ₂ = 26.60

Tableau 2.5: Application des méthodes de seuillage. Paramètres statistiques des images.

• Analyse des histogrammes :

<i>Cas</i>	<i>Méthodes</i>	<i>Taille de l'image</i>	<i>Analyse des histogrammes</i>
1-)	Isodata	170x170	- Histogramme à deux modes 0 et 255 de fréquences respectives 4721 et 24179.
2-)	Isodata	50x50	- Histogramme à deux modes 0 et 255 de fréquences respectives 693 et 1807.
3-)	Analyse discriminante	170x170	- Histogramme à deux modes 0 et 255 de fréquences respectives 10041 et 18859.
4-)	Analyse discriminante	50x50	- Histogramme à deux modes 0 et 255 de fréquences respectives 869 et 1631.

Tableau 2.6: Application des méthodes de seuillage. Analyse des histogrammes des niveaux de gris.

• Temps d'exécution et espace mémoire:

<i>Cas</i>	<i>Méthodes</i>	<i>Taille de l'image</i>	<i>Temps d'exécution (millisecondes)</i>	<i>Espace mémoire (octets)</i>
1-)	Isodata	170x170	330	30500
2-)	Isodata	50x50	50	4100
3-)	Analyse discriminante	170x170	380	30700
4-)	Analyse discriminante	50x50	60	4300

Tableau 2.7: Application des méthodes de seuillage. Temps d'exécution et espace mémoire.

2-2-3 - Conclusion:

- L'algorithme Isodata vise à affecter les pixels de l'image dans des classes distinctes, les images traitées par Isodata (Figure 2.8) et leurs histogrammes de niveaux de gris, montrent la répartition des pixels dans deux classes. Plusieurs zones homogènes ont pu être mises en évidence.

L'algorithme Isodata est basé sur le calcul des moyennes des classes, sans tenir compte d'un paramètre de dispersion, le tableau (2.5) montre que les variances des classes C_1 et C_2 obtenues par Isodata sont élevées relativement à celles obtenues par l'analyse discriminante et elles sont d'autant plus élevées que la taille de l'image est grande.

- L'application de l'algorithme de l'analyse discriminante montre (Figure 2.8) que les pixels sont répartis en deux classes, mettant ainsi, en évidence des zones homogènes sur l'image originale.

L'algorithme de l'analyse discriminante consiste à retrouver le seuil de niveau de gris maximisant la variance interclasse. Il est donc, basé sur un paramètre de dispersion.

Ceci explique le fait que les variances de chacune des deux classes obtenues sont inférieures à celles obtenues par Isodata. Aussi, une comparaison des résultats (variances des deux classes) obtenus par cet algorithme sur des images de différentes tailles, montre que les variances des deux classes sont plus faibles lorsqu'on diminue la taille de l'image.

De l'application de ces deux algorithmes, nous pouvons aussi conclure que les paramètres temps d'exécution et espace mémoire requis ne sont pas sélectifs entre les deux méthodes.

D'autre part, nous constatons aussi que la segmentation par seuillage, est meilleure si la taille de l'image est petite. Ceci nous amène, à diviser l'image en "imassettes", calculer pour chacun de ces blocs de pixels le seuil local correspondant et effectuer une segmentation par blocs. L'application suivante est une segmentation par seuillage local.

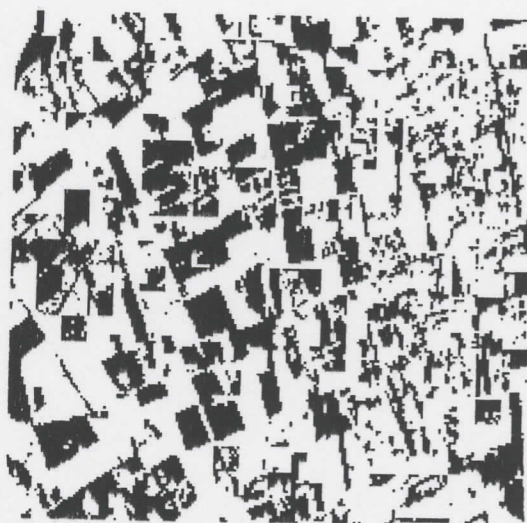
2-3- Seuillage local:

L'image est divisée en n blocs et pour chacun de ces blocs, on définit son propre seuil, par un des deux algorithmes décrits précédemment.

Application: On décompose une image originale de taille (200x200) en 400 blocs de taille 10x10, puis on applique sur chacun de ses blocs l'algorithme de seuillage Isodata.



1- Image originale 200x200
décomposée en 400 blocs de taille 10x10



2- Traitement de (1) par seuillage local (Isodata)

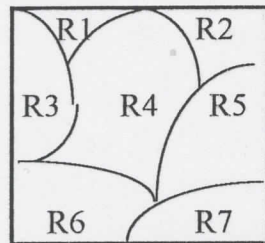
Figure 2.9 : Application du seuillage local.

Sur l'image obtenue (Figure 2.9 Image2), on constate que de nouvelles zones de l'image originale ont pu être mises en évidence, à comparer avec le seuillage global (Figure 2.8 Image2).

3- Segmentation par régions:

Contrairement aux méthodes de segmentation par détection de contours, qui s'intéressent aux frontières des objets, Les méthodes de segmentation par régions ont pour but de regrouper les points adjacents de l'image qui possèdent un attribut scalaire ou vectoriel homogène, en régions connexes. Ces méthodes s'appuient sur une coopération de tous les points appartenant à une même région et sont basées sur les propriétés intrinsèques des objets de l'image.

Le choix de ces propriétés détermine ce que l'on appelle le **critère de segmentation**. Les principaux critères utilisés sont le niveau de gris, la couleur, la texture... Dans le cas d'une image aérienne, une segmentation par la couleur permet d'isoler la végétation, les routes, les milieux aquatiques, le bâtis... Rappelons que dans notre étude, on choisit comme critère de segmentation le niveau de gris.



$$A(L,K) = \bigcup_{i=1..7} R_i$$

A: image initiale.

R_i : région de l'image.

Figure 2.10: Principe de la segmentation par régions.

Une fois avoir fixé le critère de segmentation CS, la segmentation par région consiste à vérifier le prédicat d'uniformité [52] [33]:

- 1) $P(cs, R_i) = \text{Vrai} \quad \forall i=1, \dots, M$ (M: nombre de régions)
- 2) $P(cs, R_i \cup R_j) = \text{Faux} \quad \forall i \neq j \text{ et } R_i \text{ adjacente à } R_j$

La première condition traduit l'homogénéité intérieure dans une même région, la seconde traduit l'hétérogénéité d'une zone avec les zones voisines.

3-1- Méthodes globales ou par séparation:

3-1-1- Principe de la séparation:

Initialement, l'image est divisée en quatre zones. On teste la validité du critère d'homogénéité sur chaque zone. S'il est vérifié, on passe à une autre zone, sinon on décompose la zone considérée en plus petites zones et on réitère le processus. A la fin on obtient un arbre appelé Quad_tree qui donne une structure hiérarchisée des régions: Les régions formant un noeud terminal sont homogènes, les régions formant un noeud intermédiaire sont hétérogènes, donc dérivées en quatre branches [52].

Critère d'homogénéité (Ch):

$$P(\text{Ch}, R_i) = \text{Vrai} \quad \text{Si} \quad \text{VAR}(R_i) = \frac{1}{\text{card}(R_i)} \sum_{x \in R_i} (g(x) - \mu_i)^2 < \epsilon$$

$$P(\text{Ch}, R_i) = \text{Faux} \quad \text{Sinon}$$

Où $g(x)$: valeur de niveau de gris d'un pixel x de la région R_i
 μ_i est la valeur moyenne des niveaux de gris des points de R_i
 ϵ : seuil donné .

3-1-2- Algorithme de la séparation des régions:

Fonction critère_homog(R_i);

$$\text{Variance}(R_i) = \frac{1}{\text{card}(R_i)} \sum_{x \in R_i} (g(x) - \mu_i)^2$$

Si $\text{Variance}(R_i) < \epsilon$ alors critère_homog = vrai
 sinon critère_homog = faux

Fin;

Procédure partage(R_i);

Diviser R_i en $R_i^1, R_i^2, R_i^3, R_i^4$

Si Critère_homog (R_i^1)=faux alors Partage (R_i^1);

Si Critère_homog (R_i^2)=faux alors Partage (R_i^2);

Si Critère_homog (R_i^3)=faux alors Partage (R_i^3);

Si Critère_homog (R_i^4)=faux alors Partage (R_i^4);

Fin;

Procédure principale :

Lire (ϵ);

Si Critère_homog (Image)=faux alors partage(Image)

Fin;

Algorithme 2.3: Segmentation par séparation de régions.

3-1-3- Exemple:

Soit à diviser l'image (a) en régions , le seuil ϵ est fixé à 0.2

2	2	2	2	2	2	1	1
2	2	2	2	2	2	1	1
2	2	2	2	2	2	1	1
2	2	2	2	2	2	1	1
2	2	2	1	2	2	1	1
2	2	1	1	1	1	1	1
2	2	1	1	1	1	1	1
2	1	1	1	1	1	1	1

(a) image initiale

	2	2	2	2	2	2	1	1	
R_1 →	2	2	2	2	2	2	1	1	← R_2
	2	2	2	2	2	2	1	1	
	2	2	2	2	2	2	1	1	
	2	2	2	1	2	2	1	1	
R_3 →	2	2	1	1	1	1	1	1	← R_4
	2	2	1	1	1	1	1	1	
	2	1	1	1	1	1	1	1	

(b) image découpée en 4

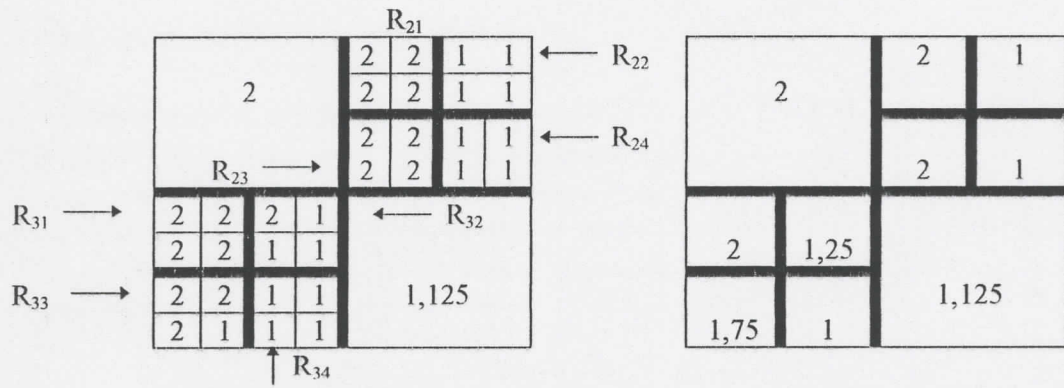


Figure 2.11: Opération de division d'une image.

Une fois l'opération de division terminée, on attribue aux pixels de chaque région la moyenne des niveaux de gris des pixels qui la composent. Le déroulement de l'opération de division sur l'image (a), est présenté dans le tableau suivant:

R_i	Moyenne de R_i	Variance de R_i	Critère d'homogénéité	Niveau de gris de R_i
R_1	2	0	vrai	2
R_2	1,5	0,25 > 0,2	Faux	/
R_{21}	2	0	vrai	2
R_{22}	1	0	vrai	1
R_{23}	2	0	vrai	2
R_{24}	1	0	vrai	1
R_3	1,5	0,25 > 0,2	Faux	/
R_{31}	2	0	vrai	2
R_{32}	1,25	0,18 < 0,2	vrai	1,25
R_{33}	1,75	0,18 < 0,2	vrai	1,75
R_{34}	1	0	vrai	1
R_4	1,125	0,109 < 0,2	Vrai	1,125

Tableau 2.8 : Déroulement de l'opération de division de l'image (a) de Figure 2.11.

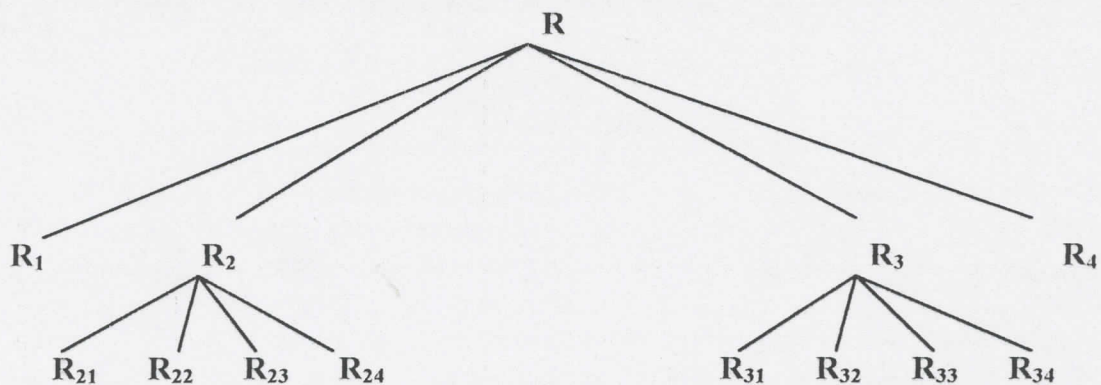


Figure 2.12- : Représentation du résultat de la division par un arbre quaternaire

3-2- Méthodes locales ou par fusion:

3-2-1- Principe de la fusion:

Cette méthode suit le principe inverse de la séparation, elle consiste à explorer l'image à partir de petites régions que l'on fusionne.

L'image issue de l'opération de division est représentée par un graphe de contiguïté des régions GCR [33]. Chaque couple de régions adjacentes est relié par une arête munie d'un poids $C(R_i, R_j)$ égal à :

$$C(R_i, R_j) = \sum_{x \in R_i \cup R_j} [g(x) - \mu(R_i, R_j)]^2 \quad (2.22)$$

avec

$$\mu(R_i, R_j) = \frac{1}{\text{card}(R_i \cup R_j)} \sum_{x \in R_i \cup R_j} g(x) \quad (2.23)$$

On détermine, dans une première phase, l'arête de poids minimum, s'il y a plusieurs, on choisit une au hasard. Puis, on réunit les deux noeuds reliés par cette arête et les n liens qui connectent ce nouveau noeud au reste du graphe sont recalculés selon l'équation (2.22). Enfin, on évalue un critère de fin de rassemblement, si ce dernier est n'est pas vérifié, on reprend le rassemblement, sinon l'opération de rassemblement est arrêté.

Critère de fin de rassemblement (Cfr):

Le critère de fin de rassemblement est tel que la somme des erreurs quadratiques (SEQ) des niveaux de gris de toute l'image excède un seuil donné ϵ .

$$P(Cfr, R_i) = \text{Vrai} \quad \text{Si} \quad \text{SEQ}(m) = \sum_{i=1}^m \sum_{x \in R_i} (g(x) - \mu(R_i))^2 > \epsilon \quad (\epsilon: \text{seuil donné})$$

$$P(Cfr, R_i) = \text{Faux} \quad \text{Sinon}$$

3-2-2- Algorithme de la fusion:

Etape 1: Chercher l'arête de poids minimum, s'il y a plusieurs, choisir une au hasard.

Etape 2: Fusionner les noeuds extrémités de cette arête, en un seul noeud.

Recalculer les poids des connexions de ce nouveau noeud au reste du graphe

Evaluer le critère de fin de rassemblement :

$$\text{SEQ}(m) = \sum_{i=1}^m \sum_{x \in R_i} (g(x) - \mu(R_i))^2$$

Où - m est le nombre de régions= nombre de noeuds du graphe.

- $g(x)$ est le niveau de gris du pixel x .

Si $\text{SEQ}(m) > \epsilon$

Alors Fin

Sinon aller à étape 1.

Fsi;

Fin;

Algorithme 2.4: Fusion des régions.

3-2-3- Exemple:

Nous déroulons, l'algorithme (2.4) sur l'image (d - Figure 2.11) obtenue par une opération de division et représentée par son graphe de contiguïté.

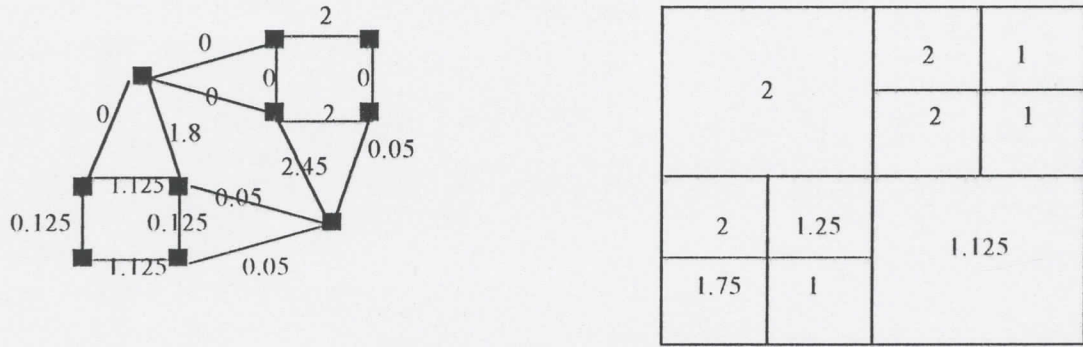


Image et moyennes des régions

Figure 2.13 : Graphe de contiguïté des régions de l'image (d) de Figure 2.11.

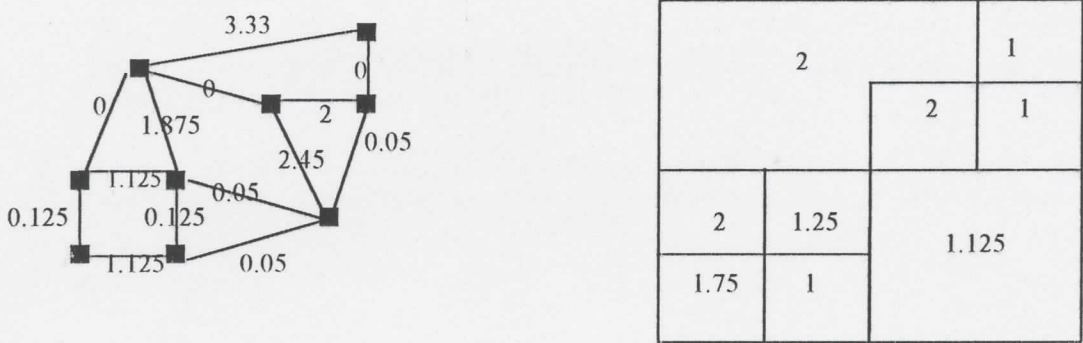


Image après fusion de deux régions

Figure 2.14 : Première itération de l'opération de fusion. Graphe de contiguïté des régions. Image résultat.

3-3- Méthodes par séparation/fusion:

C'est la combinaison des deux méthodes précédentes, la segmentation est effectuée en deux phases:

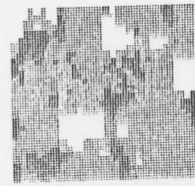
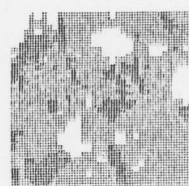
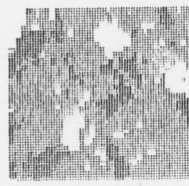
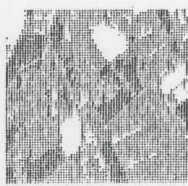
La première est descendante: séparation

La deuxième est ascendante: fusion; l'arbre quad-tree obtenu à la fin de la première phase est parcouru dans le sens feuilles-racine, on fusionne au fur et à mesure les régions vérifiant le critère de similarité.

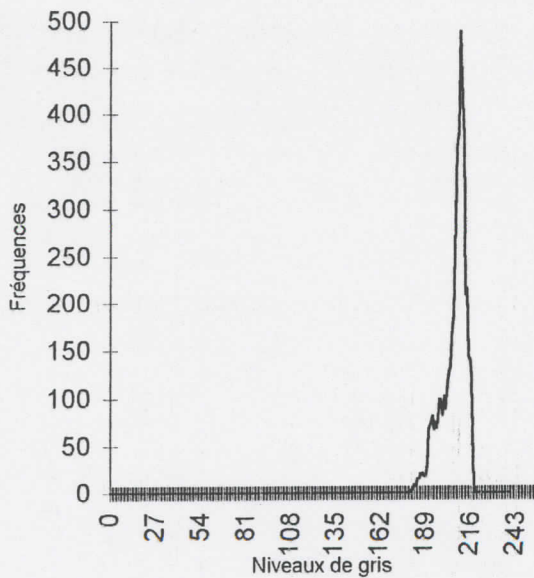
3-4- Application de la segmentation par régions:

Nous appliquons sur une image satellite de la région de l'Atlas Blidéen, prise par le satellite SPOT, la méthode de segmentation par régions. En raison du problème de débordement de la pile du système (STACK) lié aux appels récursifs de la procédure de partage de l'image, quelques restrictions sur la taille de l'image s'imposent: celle-ci est sous échantillonnée en fenêtres de taille 70x70 pixels. La procédure est testée pour trois valeurs du seuil: $\epsilon = 1$, $\epsilon = 6$ et $\epsilon = 10$.

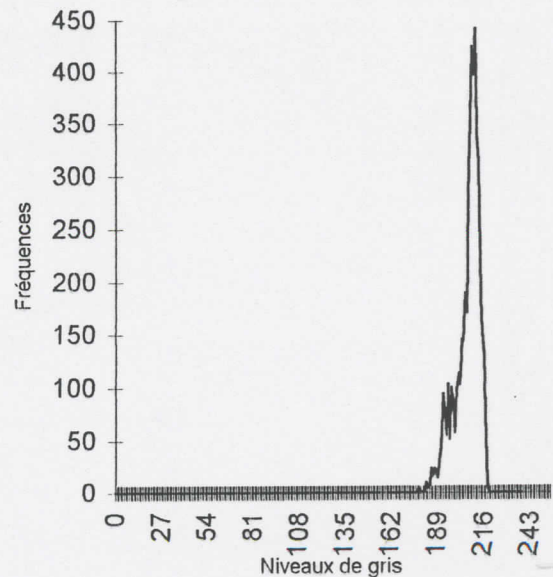
3-4-1- Images et histogrammes des niveaux de gris:



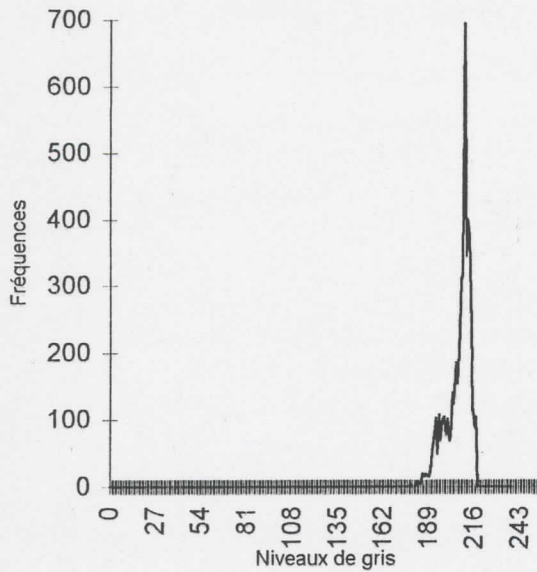
1-) Image originale

2-) Application avec $\epsilon = 1$ 3-) Application avec $\epsilon = 6$ 4-) Application avec $\epsilon = 10$ 

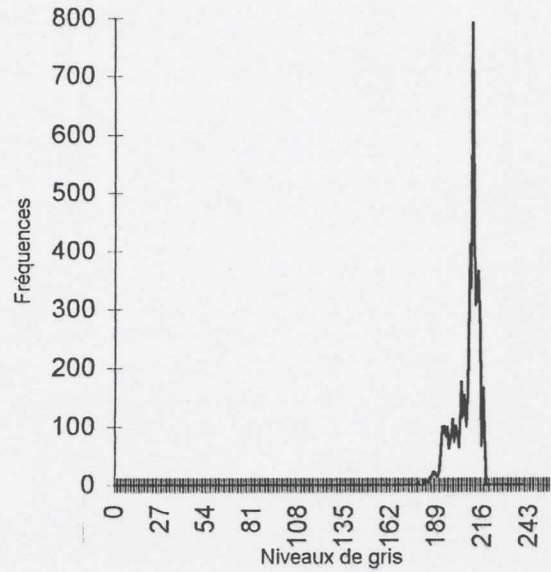
5-) Histogramme de (1)



6-) Histogramme de (2)



7-) Histogramme de (3)



8-) Histogramme de (4)

Figure 2.15: Application de la segmentation par régions. (1-4) : Images ; (5-8) Histogrammes des niveaux de gris.

3-4-2- Analyse comparative:

Cette analyse est basée sur l'appréciation visuelle des images, leurs histogrammes ainsi que l'espace mémoire et le temps d'exécution nécessaires pour les différentes valeurs du seuil de la variance.

- Analyse visuelle des images :

<i>Paramètres de la méthode</i>	<i>Analyse visuelle</i>
$\epsilon = 1$	- Dilatation des zones de niveau de gris extrémal
$\epsilon = 6$	- Dilatation plus importante avec disparition des points isolés
$\epsilon = 10$	- Disparition des points isolés - Dilatation des régions fermées par rapport aux points isolés.

Tableau 2.9: Application de la segmentation par régions. Analyse visuelle.



- Analyse des histogrammes des niveaux de gris :

<i>Paramètres de la méthode</i>	<i>Analyse des histogrammes</i>
$\varepsilon = 1$	- Apparition d'un deuxième mode.
$\varepsilon = 6$	- Elargissement de la nouvelle classe de l'histogramme.
$\varepsilon = 10$	- Distinction des deux modes de l'histogramme.

Tableau 2.10: Application de la segmentation par régions.
Analyse des histogrammes

- Temps d'exécution et espace mémoire:

<i>Paramètres de la méthode</i>	<i>Temps d'exécution (millisecondes)</i>	<i>Espace mémoire (octets)</i>
$\varepsilon = 1$	3700	74100
$\varepsilon = 6$	1528	29000
$\varepsilon = 10$	1400	25200

Tableau 2.11: Application de la segmentation par régions.
Temps d'exécution et espace mémoire.

3-4-3- Conclusion:

Par l'application de la segmentation par régions sur une image satellitaire spot, nous obtenons une image avec des dilatations locales des zones de niveau de gris homogène. Ceci est dû au fait que les images satellitaires, présentent des variances locales très élevées, par rapport à la variance globale de toute l'image. Nous concluons, alors que la segmentation par régions, de part sa conception basée essentiellement sur le paramètre variance, ne fournit pas de bons résultats.

4- Méthodes Algébriques:

Les méthodes algébriques s'appuient sur les caractéristiques topologiques de l'image et sont de deux types: les méthodes basées sur la morphologie mathématique et les méthodes basées sur la prétopologie mathématique.

4-1- La Morphologie Mathématique:

Les opérateurs morphologiques reposent sur le concept de transformation géométrique d'une image par un élément structurant. L'élément structurant est un masque caractérisé par son point central et par son motif [46]:

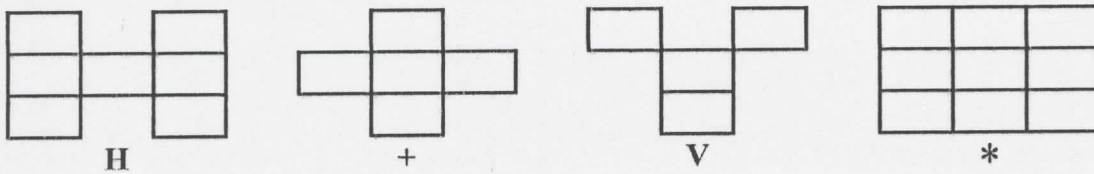


Figure 2.16 :Types d'éléments structurants.

L'application de l'opérateur morphologique consiste à balayer l'image avec ce masque. Son point central est superposé avec le pixel courant. On effectue alors une opération d'ensemble entre le motif du masque et la partie de l'image qui lui est correspondante. Ces opérations sont: l'union (OR), l'intersection (AND), la complémentation (NOT) et la différence symétrique(XOR):

Exemple:

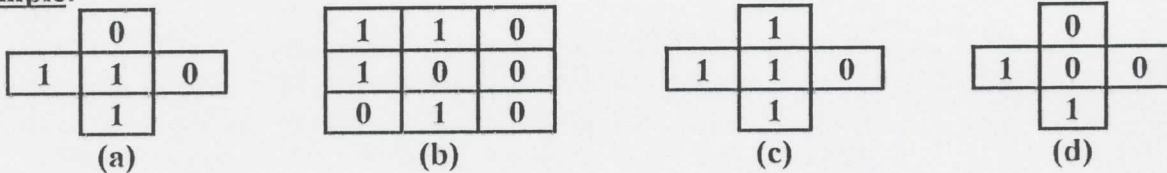


Figure 2.17: Application d'un élément structurant.

(a):Masque (élément structurant en +). (b):Fenêtre 3x3 sur l'image originale.
 (c):Résultat de l'union de (a) et (b). (d): Résultat de l'Intersection de (a) et (b).

Dans notre étude, on applique les opérateurs morphologiques sur des images binaires. La binarisation d'une image consiste à transformer cette dernière en une image présentant deux niveaux de gris seulement, généralement le niveau 0 et 255. On attribue aux pixels de la première classe une valeur '0' et ceux de la deuxième classe une valeur '1'.

4-1-1- Dilatation:

Dans le cas de binarisation d'une image fortement bruitée, il apparaît sur l'image des points noirs isolés au milieu des parties blanches et n'appartenant pas à la scène mais qui sont dus au bruit. La dilatation permet d'éliminer ces points. Elle consiste à réaliser pour

chaque pixel le « OR » logique de ses voisins. Si le résultat est à « 1 » alors le niveau de gris du point central est forcé à 1 dans l'image résultat, sinon sa valeur est conservée [52].

Exemple:

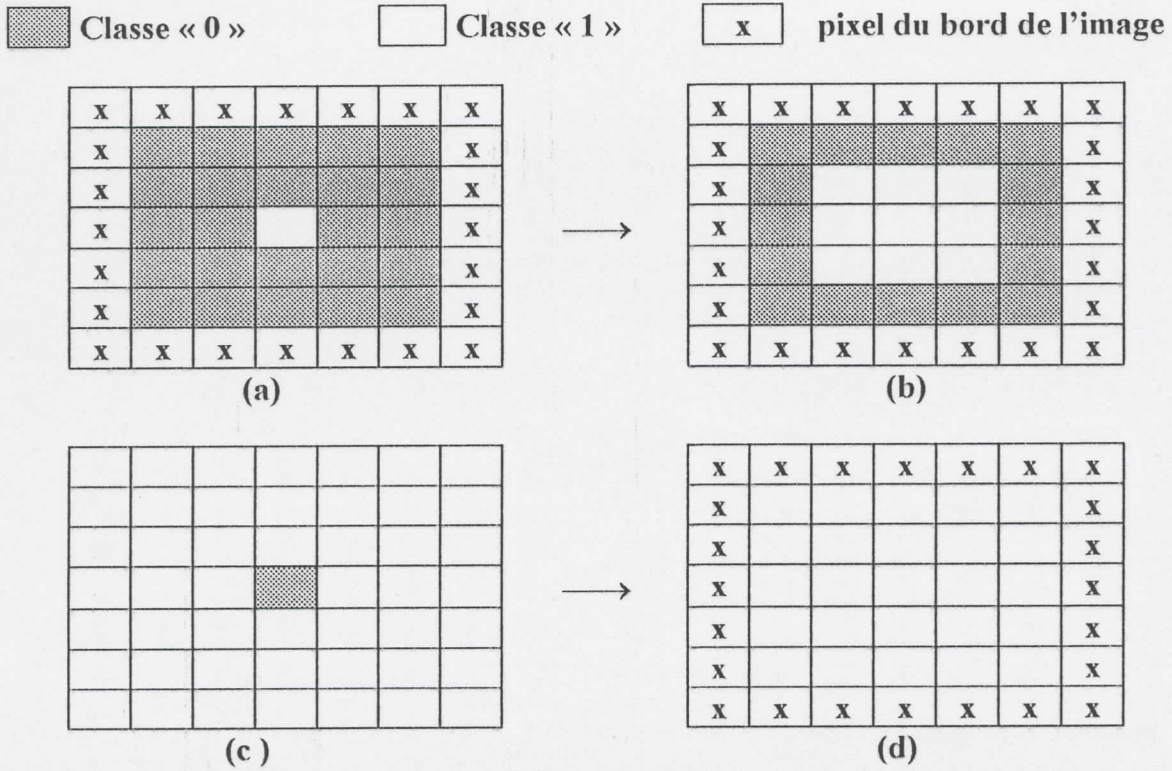
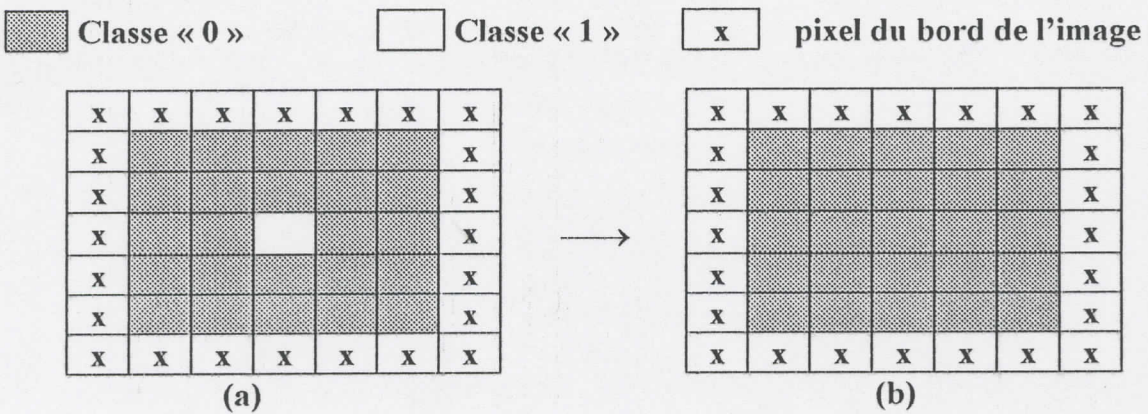


Figure 2.18: Application de la dilatation.
 a-): image originale. b-): Résultat de la dilatation sur (a)
 c-): image originale. d-): Résultat de la dilatation sur (c).

4-1-2- Erosion:

C'est l'opération duale de la dilatation. Elle élimine les points isolés blancs se trouvant au milieu des parties noires. Ceci est réalisé en effectuant pour chaque pixel le « ET » logique de ses voisins. Si le résultat est à « 1 », la valeur du niveau de gris du pixel central est conservée, sinon elle est forcée à zéro [52].

Exemple:



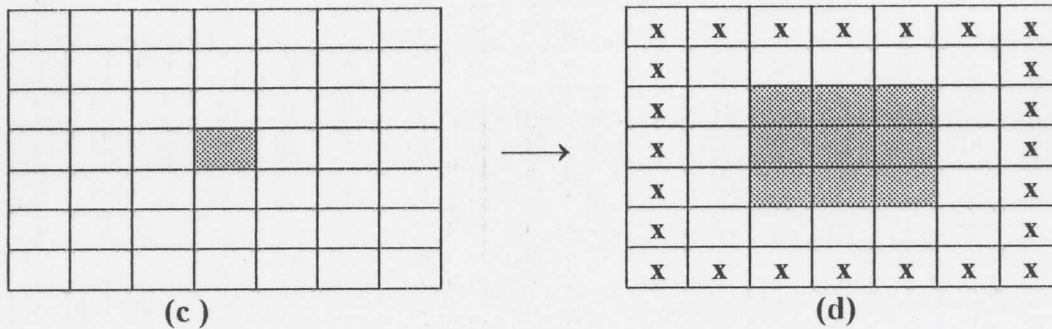


Figure 2.19: Application de l'érosion.
 a-): image originale. b-): Résultat de l'érosion sur (a)
 c-): image originale. d-): Résultat de l'érosion sur (c).

4-1-3- Squelettisation:

La squelettisation d'une image binaire permet d'extraire son squelette. Celui-ci représente la quantité d'informations minimale permettant de décrire complètement l'image. Le squelette d'une image binaire s'obtient en réalisant une opération XOR entre l'image originale et son érodée.

Exemple:

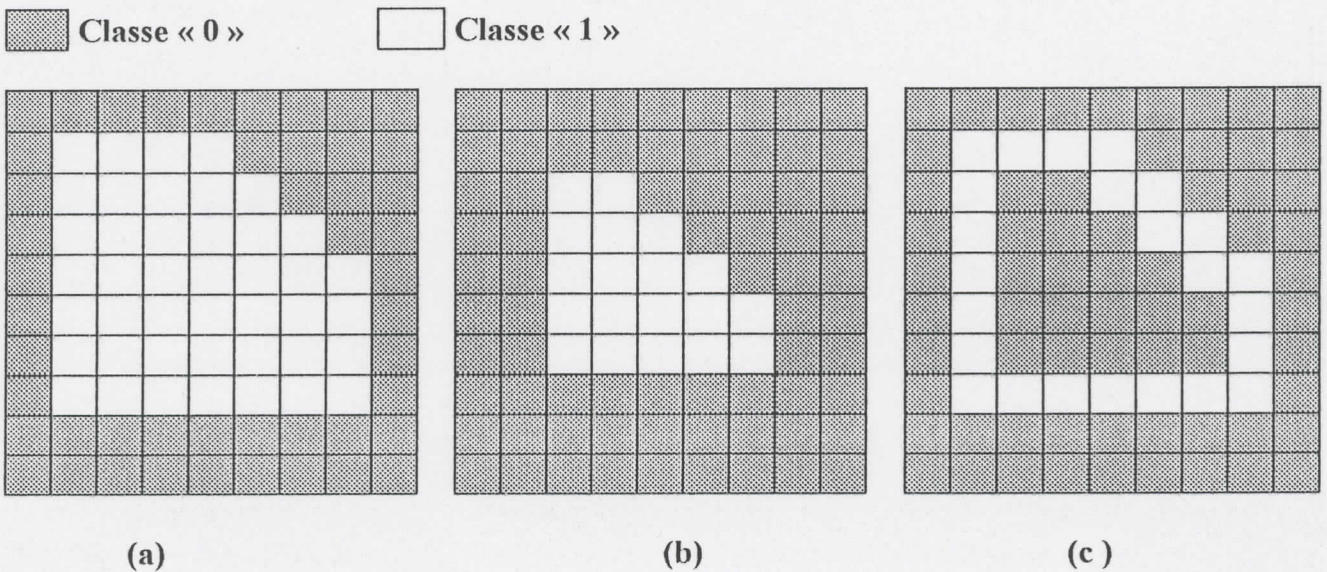
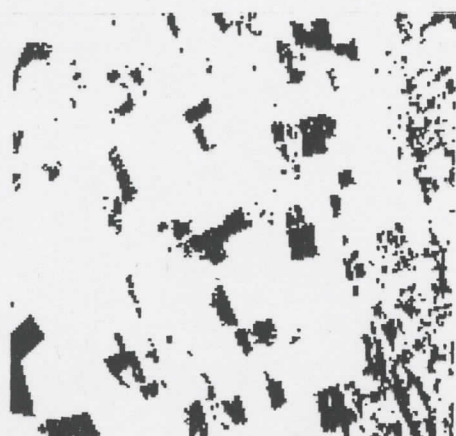


Figure 2.20: Application de la squelettisation.
 a-) : Image originale. b-): Résultat de l'érosion sur (a). c-): Résultat de (a) XOR (b).

4-2- Application de la morphologie mathématique:

Les opérations de dilatation, érosion et squelettisation sont appliquées sur une image binaire obtenue en utilisant la méthode de seuillage Isodata présentée au (§ 2-1-1 p. 34).

4-2-1- Affichage des Images:



a-) Image originale binaire



b-) Dilatation de (a)



c-) Erosion de (a)



d-) Squelettisation de (a)

Figure 2.21: Application des opérateurs morphologiques.

a-) : Image binaire. b-): Image diluée de (a) . c-): Image érodée de (a). d-): Squelette de (a)

4-2-2- Analyse comparative:

Nous comparons les caractéristiques des opérateurs morphologiques, selon les images obtenues ainsi que, le temps d'exécution et l'espace mémoire nécessaires à chaque méthode.

- Analyse visuelle des images:

<i>Méthode</i>	<i>Analyse visuelle</i>
Dilatation	- Rétrécissement des zones noires de l'image au profit des zones blanches et élimination des pixels noirs isolés.
Erosion	- Elargissement des zones noires de l'image.
Squelettisation	- Extraction des contours de l'image.

Tableau 2.12: Application des opérateurs morphologiques. Analyse visuelle.

• **Temps d'exécution et Espace mémoire:**

Méthode	Temps d'exécution (millisecondes)	Espace mémoire (octets)
Dilatation	110	29000
Erosion	50	29000
Squelettisation	60	30100

Tableau 2.13: Application des opérateurs morphologiques. Temps d'exécution et espace mémoire.

4-2-3- Conclusion:

La dilatation et l'érosion améliorent l'apparence de l'image binaire, en éliminant respectivement les pixels noirs et les pixels blancs isolés. La squelettisation permet plutôt d'extraire tous les contours de l'image binaire.

Ces méthodes reposent sur des opérations logiques, le temps d'exécution et l'espace mémoire requis par chaque méthode sont négligeables. Ces deux paramètres dépendent uniquement de la taille de l'image à traiter. La squelettisation présente un temps plus élevé car elle se subdivise en deux opérations élémentaires qui sont l'érosion et la disjonction exclusive.

4- 3- La prétopologie mathématique:

La méthode que nous présentons, repose sur la forme ensembliste de l'image. Chaque objet de l'image est un ensemble de points voisins ayant des propriétés communes. La détermination de ces ensembles utilise une technique reposant sur la prétopologie mathématique. Cette méthode proposée par N. Selmaoui [48], est un groupement par propagation reposant sur la définition d'une structure prétopologique sur l'ensemble à classifier.

4-3-1- Généralités sur les structures prétopologiques [48][26]:

1) Définitions:

Soit A un ensemble non vide ($A \neq \emptyset$).

Soit A' un sous ensemble de A ($A' \subset A$).

Soit a un élément de A ($a \in A$).

On note: $P(A)$ l'ensemble des parties de A.

$C_{A'}^A$ le complémentaire de A' dans A ($C_{A'}^A \cup A' = A$ et $C_{A'}^A \cap A' = \emptyset$)

- a ∈ A est adhérent à A', s'il existe des éléments de A' « aussi proche que l'on veut de a » ie:

$$\forall \epsilon > 0, \exists x \in A' / d(a,x) = \| a - x \| < \epsilon$$

• On appelle adhérence ou fermeture ou prétopologie sur A toute application de P(A) dans P(A) notée ad_A vérifiant:

- 1) $ad_A(\emptyset) = \emptyset$
- 2) $\forall A' \in P(A), A' \subset ad_A(A')$

• Le couple (A, ad_A) est appelé espace prétopologique

• On appelle intérieur de A' la partie notée $INT(A')$ définie par:

$$\forall A' \in P(A) \quad int(A') = C^{\wedge}(ad_A(C^{\wedge}(A')))$$

• Soit (A, ad_A) un espace prétopologique:

A' est fermé si $ad_A(A') = A'$

A' est ouvert si $int(A') = A'$

2) Comparaison entre prétopologie et ordre :

Soit $P_A = \{ad_{1A}, ad_{2A}, \dots\}$ l'ensemble des prétopologies définies sur A. On définit sur P_A , la relation d'ordre notée " \ll " par:

$$ad_{2A} \ll ad_{1A} \Leftrightarrow \forall A' \in P(A) \quad ad_{1A}(A') \subset ad_{2A}(A')$$

On dit que ad_2 est moins fine que ad_1

1) P_A est partiellement ordonné par la relation d'ordre " \ll "

2) P_A possède pour la relation " \ll ", un plus petit élément appelé **topologie grossière** noté ad_g et un plus grand élément noté ad_d appelé **topologie discrète**.

3) Composition de prétopologie :

On munit l'ensemble P_A des prétopologies de A, d'une loi de composition notée O définie par:

$$\begin{aligned} O: P_A \times P_A &\longrightarrow P_A \\ (ad_1, ad_2) &\longrightarrow ad_{12} = ad_1 \circ ad_2 = ad_1(ad_2(A')) \end{aligned}$$

4) Construction de prétopologies :

Soit R une relation binaire définie sur A.

- On désigne par Γ la multiapplication de A dans P(A) définie par:

$$\forall x \in A, \Gamma(x) = \{y \in A / y R x\}$$

- On désigne par Γ^{-1} la multiapplication de A dans P(A) définie par:

$$\forall x \in A, \Gamma^{-1}(x) = \{y \in A / x R y\}$$

- On définit à partir de Γ et Γ^{-1} les multiapplications suivantes:

$$\Gamma^p = \Gamma(\Gamma^{p-1}(x)) \quad p \geq 1$$

$$\Gamma^{-p} = \Gamma^{-1}(\Gamma^{-p+1}(x)) \quad p \geq 1$$

$$\text{avec } \Gamma^0(x) = \{x\}$$

$$\Gamma^- = \Gamma(x) \cup \{x\}$$

$$\Gamma^{-1-} = \Gamma^{-1}(x) \cup \{x\}$$

5) Prétopologie des k-plus proches voisins:

Soit k un entier naturel non nul, soit R_k la relation binaire sur A définie par:

$$\forall x \in A, \forall y \in A, x R_k y \Leftrightarrow \{ y \text{ est l'un des } k \text{ plus proches voisins de } x \text{ dans } A \}$$

On considère les multiapplications Γ et Γ^{-1} définies par :

$$\begin{aligned} \forall x \in A, \Gamma(x) &= \{x\} \cup \{y \in A / y R_k x\} \\ \forall x \in A, \Gamma^{-1}(x) &= \{x\} \cup \{y \in A / x R_k y\} \end{aligned}$$

Ces deux multiapplications engendrent les prétopologies des k-plus proches voisins et des k-plus proches voisins réciproques notées $V(k,.)$ et $V^{-1}(k,.)$. elles sont définies par:

$$\begin{aligned} V(k, \emptyset) &= V^{-1}(k, \emptyset) = \emptyset \\ \forall A' \in P(A), V(k, A') &= \{y \in A / \Gamma^{-1}(y) \cap A' \neq \emptyset\} = \bigcup_{x \in A'} \Gamma(x) = \bigcup_{x \in A'} V(k, \{x\}) \\ \forall A' \in P(A), V^{-1}(k, A') &= \{y \in A / \Gamma(y) \cap A' \neq \emptyset\} = \bigcup_{x \in A'} \Gamma^{-1}(x) = \bigcup_{x \in A'} V^{-1}(k, \{x\}) \end{aligned}$$

4-3-2- Description de la méthode :

Soient A : la matrice image.

g: fonction des niveaux de gris de l'image.

$A' = V_8(p)$: ensemble des 8 voisins, ($A' \subset A$)

On désigne par V, la fonction de voisinage d'un point p de l'image:

$$V(p) \subset V_8(p) \cup \{p\}$$

Soit R_1 la relation binaire telle que:

$$\forall p, q \in A, p R_1 q \Leftrightarrow g(p) \leq g(q)$$

Soit Γ_1 la multiapplication définie sur l'ensemble image A par:

$$\forall p \in A, \Gamma_1(p) = \{q \in A' / q R_1 p\}$$

Soit ad_1 l'adhérence associée à Γ_1 définie par:

$$ad_1(A') = \{p \in C_{A'}^A / \forall q \in V(p) \cap A', q \in \Gamma_1(p)\} \cup A'$$

4-3-3-Principe de la méthode:

Cette méthode repose sur le groupement par propagation, en construisant des classes par calcul d'adhérences successives de points idéaux, qui dans ce cas seront les points minima locaux de la fonction niveau de gris. Un point appartenant en même temps à deux classes différentes est un point de contour:

- Dans un premier temps, on détermine les points idéaux: points minima locaux de la fonction de gris.
- On calcule pour chacun de ces points son adhérence, si un point appartient en même temps à deux classes différentes, il sera qualifié de point appartenant à un contour, en conséquence, il ne sera pas pris en considération à la prochaine itération.
- Le processus s'arrête si la stabilité pour chacune des classes est atteinte.

4-3-4- Algorithme de la Segmentation par prétopologie [48]:

C désignera l'ensemble des points contours

1. Etape initiale

$i \leftarrow 0;$

(rechercher tous les points minima locaux et les marquer comme appartenant aux classes A_i^0 qui sont vides au départ; la variable i permet de fournir le nombre de classes déterminées)

Pour chaque point p de l'image **Faire**

Si p est minimum local

Alors $i \leftarrow i+1$

$A_i^0 \leftarrow \{p\}$

Fin si

Fin pour

$m \leftarrow i$ (nombre de classes)

2. Etape intermédiaire

(regrouper les points minima locaux qui sont voisins en une classe: ainsi on débutera le processus par une zone minimale)

Pour tout couple $(i,j), i < j$ **Faire**

Si $p \in A_i^0$ et $q \in A_j^0$ et p et q sont voisins

Alors mettre q dans A_i^0

Pour tout $k > i+1$ **Faire**

$A_{k-1}^0 \leftarrow A_k^0$

Fin pour

Décrémenter les indices $m \leftarrow m-1$

Fin si

Fin pour

3. Etape de construction (itérative)**3.1. Pour** $i \leftarrow 1$ à m **Faire**

(mettre à jour la fonction de voisinage qu'il faut utiliser pour calculer les nouvelles adhérences et ceci en neutralisant tous les points déjà classés dans C)

Pour tout point $p \in A_i^k$ **Faire**

$V(p) \leftarrow (V_8(p) \cup \{p\}) \setminus C$

Fin pour

Calculer $A_i^{k+1} = \text{ad}_1(A_i^k)$

Fin pour

3.2. Pour tout couple $(i,j), i \neq j$ **Faire**

$C \leftarrow C \cup (A_i^{k+1} \cap A_j^{k+1})$

(mettre tous les points de l'intersection i.e. les points contours dans C)

3.3. Si $\exists i$ tel que $A_i^{k+1} \neq A_i^k$ **Alors** aller en **3.1.**

Si $\forall i, A_i^{k+1} = A_i^k$ **Alors** aller en **3.4.**

(On ne s'arrête que si toutes les classes sont fermées)

3.4. Arrêt

Algorithme 2.5: Segmentation par prétopologie selon N. SELMAOUI.

4-4- Application de la prétopologie mathématique:

La segmentation par prétopologie mathématique est appliquée sur une fenêtre de taille 100x100 d'une image satellite SPOT XS1, représentant la région de l'Atlas Blidéen. En raison de la structure de données utilisée dans cette procédure, la taille maximale de l'image à traiter est de 100x100.

4-4-1- Images et histogrammes des niveaux de gris:

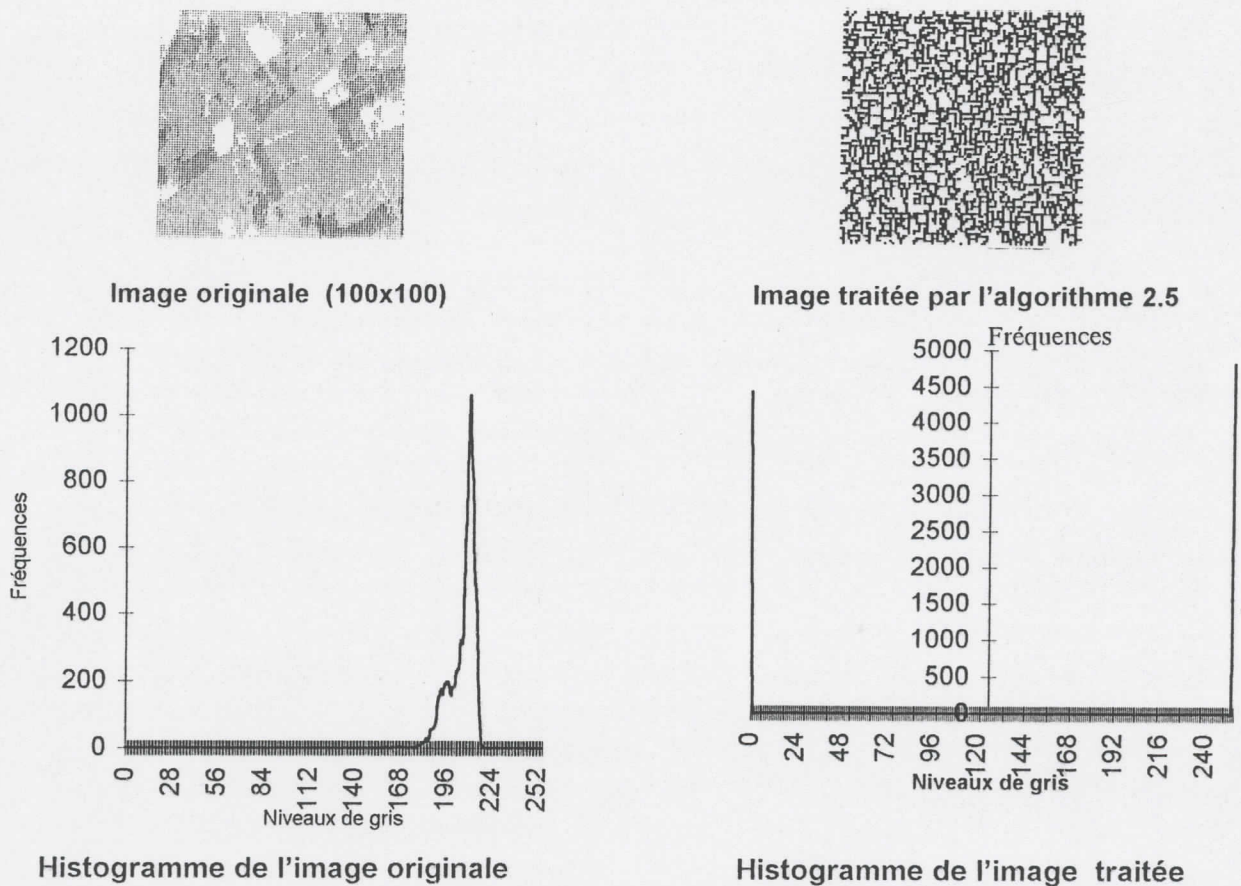


Figure 2.22: Application de l'algorithme 2.5: Images et Histogrammes.

4-4-2- Analyse des résultats:

- Analyse visuelle des images :

<i>Méthode</i>	<i>Analyse visuelle</i>
- Prétopologie	- Apparition des lignes de contours.

Tableau 2.14: Application de la prétopologie. Analyse visuelle.

- Analyse de l'histogramme :

<i>Méthode</i>	<i>Analyse d' histogramme</i>
- Prétopologie	- Histogramme bimodal, représentant les deux classes contours et intérieur des régions.

Tableau 2.15: Application de la prétopologie.
Analyse de l'histogramme.

- Temps d'exécution et espace mémoire:

<i>Méthode</i>	<i>Temps d'exécution (ms)</i>	<i>Nombre d'itérations</i>	<i>Espace mémoire (octets)</i>
- Prétopologie	5000	10	97850

Tableau 2.16: Application de la prétopologie.
Temps d'exécution et espace mémoire.

4-4-3- Discussion des résultats:

L'image obtenue par l'application de l'algorithme 2.5, présente tous les contours correspondant à la moindre transition des niveaux de gris sur l'image originale. L'histogramme des niveaux de gris présente deux pics correspondants à deux classes de niveaux de gris, les contours et les régions. Ces deux classes regroupent les pixels de l'image en nombre presque égal, ce qui est contraire au principe de la détection des contours.

Nous proposons de rajouter à cet algorithme une procédure d'élimination des contours négligeables de faible transition.

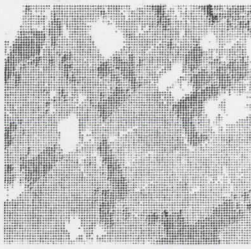
4-4-4- Procédure de sélection de contours:

Sur l'image obtenue par l'application de l'algorithme 2.5, les pixels appartenant à des classes minimales sont sélectionnés selon une procédure de seuillage des transitions entre deux pixels adjacents. Un pixel appartenant à une classe minimale est affecté à la classe des contours si l'équation (2.24) est satisfaite:

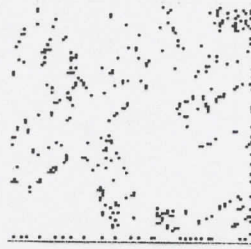
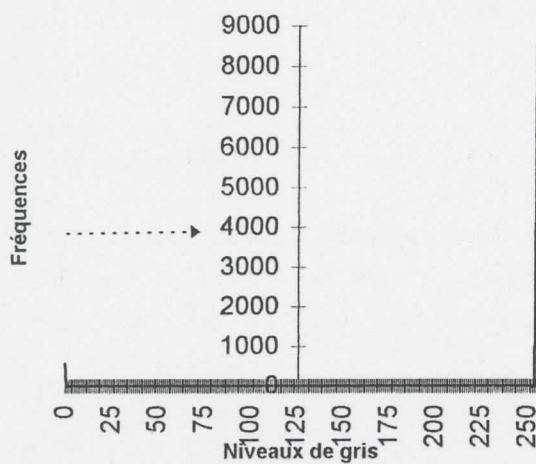
$$\text{abs}(G[i,j]-G[i,j+1]) + \text{abs}(G[i,j]-G[i+1,j]) > \varepsilon \quad (2.24)$$

La figure suivante montre des images obtenues par l'algorithme 2.5 et dont les contours négligeables ont été éliminés.

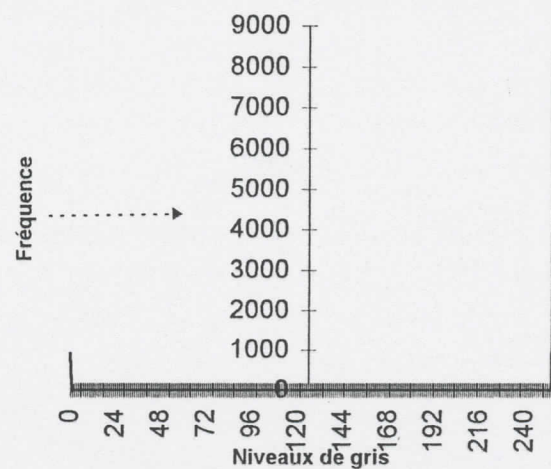
- **Images et histogrammes des niveaux de gris:**



1-) Image originale (100x100)

2-) Image traitée par l'algorithme 2.5 et élimination des contours négligeables avec $\epsilon=10$ 3-) Image traitée par l'algorithme 2.5 et élimination des contours négligeables avec $\epsilon=6$ 

Histogramme de l'image (2)



Histogramme de l'image (3)

Figure 2.23: Elimination des contours négligeables: Images et Histogrammes.

4-4-5- Conclusion:

La segmentation par prétopologie mathématique est basée sur la propagation des régions et la sélection des pixels appartenant à l'intersection de deux régions adjacentes. Une telle approche permet la détection de la plus part des transitions de niveaux de gris. Un seuillage sur ces transitions permet d'éliminer les contours négligeables. La figure (2.23) illustre la procédure de sélection des contours. Nous constatons d'après cette figure que la mise en évidence des contours des régions de l'image est paramétrée par le seuil ϵ . D'après les différentes applications, que nous avons effectuées, une valeur de $\epsilon = 6$, fournit de bons résultats en segmentation.

CHAPITRE 3 :

L'APPROCHE ALEATOIRE DU RECUIT SIMULE EN SEGMENTATION D'IMAGES

- 1- Présentation de l'Approche.
- 2- Recuit Simulé en Segmentation des Images.
 - 2-1- Recuit1.
 - 2-2- Recuit2.
 - 2-3- Recuit3.

Chapitre 3: *APPROCHE ALEATOIRE DU RECUIT SIMULE EN SEGMENTATION D'IMAGES*

INTRODUCTION

Dans le chapitre précédent, les méthodes de segmentation développées sont pour la plus part des méthodes directes (filtrage, seuillage) ou itératives mais déterministes (prétopologie et la segmentation par régions). Ce chapitre est consacré à l'approche aléatoire, où l'affectation des éléments de l'image à une classe se fait par l'utilisation de vecteurs probabilistes et où chacun des vecteurs évolue au cours d'itérations et converge vers une classification correcte [43].

Parmi les méthodes aléatoires, le recuit simulé a été introduit par Kirkpatrick et al en 1983 [31]. Cette idée prend ses origines de l'évolution des phénomènes physiques liés à la thermodynamique.

1- PRESENTATION DE L'APPROCHE :

1-1- Le Recuit Thermique:

Le **recuit thermique** est un processus physique généralement utilisé en métallurgie afin d'améliorer les qualités mécaniques d'un solide, en cherchant son état d'énergie minimale [15]. Le solide, porté à une grande température, est conduit à la phase de fusion. Les particules qui le composent présentent alors un état énergétique donné. Son refroidissement l'amènera à des états de basse énergie correspondant à de nouvelles configurations (dispositions) des particules qui le composent [18].

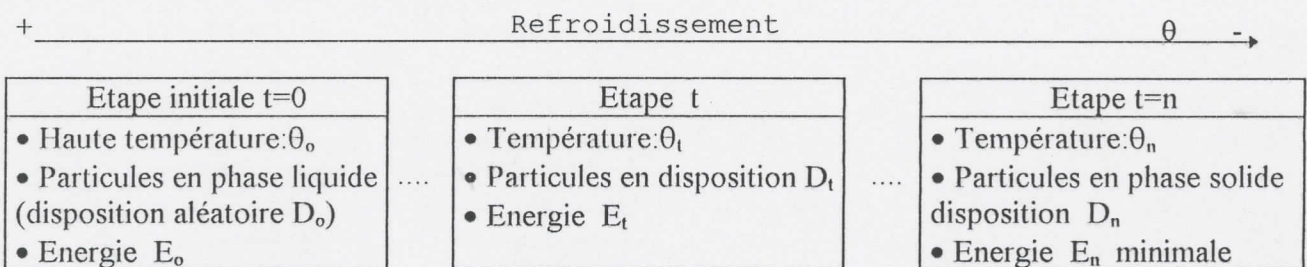


Figure 3.1: Phénomène de refroidissement d'un système physique.

DeCarvalho [15] explique le phénomène de solidification en métallurgie:

A chaque température θ_t , le métal est mené à l'équilibre thermodynamique caractérisé par un état énergétique E_t , suivant la distribution de probabilité de Boltzman [15]:

$$P(\text{Energie} = E_t) = \frac{1}{Z(\theta_t)} \text{Exp} \left(\frac{-E_t}{C_B \theta_t} \right) \quad (3.1)$$

Où $Z(\theta_t)$: Facteur de normalisation.

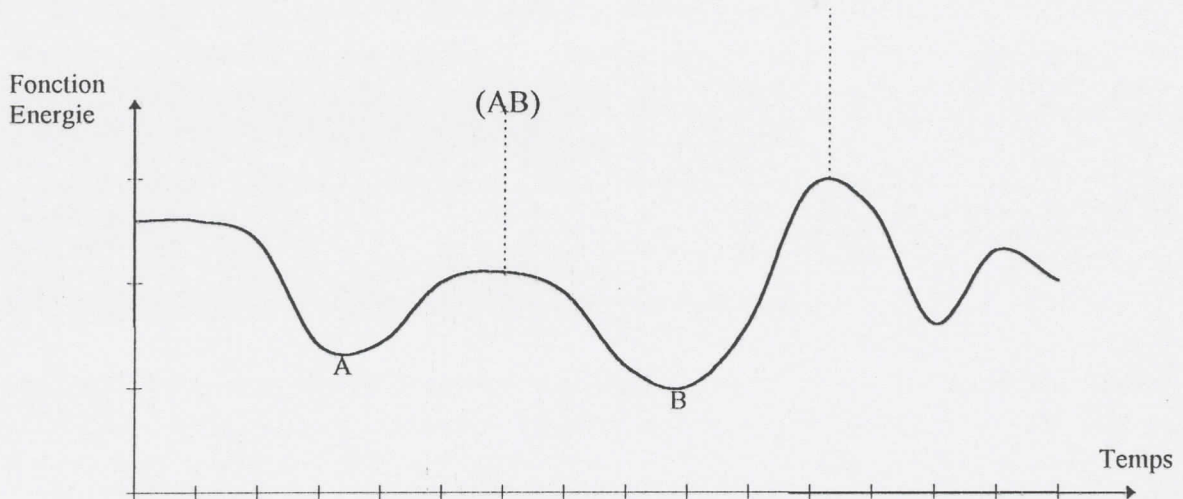
C_B : constante de Boltzman.

$\text{Exp} \left(\frac{-E_t}{C_B \theta_t} \right)$: Facteur de Boltzman.

Lorsque l'équilibre thermodynamique est atteint, on diminue la température et on attend que l'équilibre thermodynamique s'établisse à la nouvelle température. Et au fur et à mesure que la température baisse, la distribution de Boltzman se concentre sur les états de plus basse énergie conformément aux lois de la thermodynamique[15].

Cependant l'existence d'états métastables peut piéger le système à basse température, dans l'un de ces états. Il faudrait que la température soit assez élevée pour le faire sortir d'un tel état, et pas trop grande afin d'assurer le refroidissement qui l'amènera à un état stable [18] [11].

Donc lors de son refroidissement, le système converge vers un état stable (énergie minimale) et ceci est étroitement lié à la vitesse de décroissance de la température. La décroissance lente de la température organise convenablement le système cristallographique solide [11].



A: Minimum local de la fonction Energie (état métastable).

B: Minimum global de la fonction Energie (état stable).

Figure 3.2: Paysage d'Énergie.

La figure (3.2) est un exemple d'une courbe de variation d'une fonction énergie. Si le système est piégé dans le puits d'énergie "A", il faudrait que la température soit assez grande, pour lui faire sauter la barrière d'énergie (AB), puis doit diminuer afin qu'il soit attiré vers le point B (minimum global de la fonction énergie).

• Algorithme de Metropolis:

Metropolis et al [37] ont proposé un algorithme simulant l'évolution d'un solide:

- Etant donné l'état courant du solide (disposition de ses particules) d'énergie E_1 . Une faible perturbation est apportée à cet état, produisant ainsi un nouvel état d'énergie E_2 .
- Si la perturbation donne un état d'énergie plus bas, le processus remplace l'état courant par le nouvel état et réitère.
- Si la perturbation donne un état d'énergie supérieure ou égale, alors la probabilité d'acceptation du nouvel état est donné par:

$$\text{Exp} \left(\frac{-\Delta E}{C_B \theta} \right) \quad (3.2)$$

Après un grand nombre de perturbations, la distribution des probabilités des états s'approche de la distribution de Boltzman et le système est en équilibre [15].

1-2- Le Recuit Simulé:

Faisant suite aux travaux de Metropolis et 30 années plus tard Kirkpatrick et al [31], simulent le problème de minimisation de la fonction objective d'un problème d'optimisation combinatoire, par le processus de refroidissement d'un solide d'où l'approche du **recuit simulé**.

"L'idée de Kirkpatrick est de combiner les avantages de la dynamique haute température (évolution libre et rapide dans l'espace des états à la recherche de l'état minimal) et de la dynamique basse température (dynamique lente et contrainte piégeable dans les puits d'énergie)" [11].

On peut définir un problème d'optimisation combinatoire à l'aide du triplet (Ω, \mathcal{X}, H) :

Où Ω : ensemble de N éléments appelés sites.

\mathcal{X} : ensemble fini de configurations du système Ω .

H: une application $H: \mathcal{X} \rightarrow \mathbb{R}$ dite fonction objective qui à chaque configuration x de \mathcal{X} associe un nombre réel.

La recherche du minimum x_{\min} de la fonction H (ie. $x_{\min} \in \mathcal{X}$ tel que $H(x_{\min}) = \text{Min}(H(x))$), consiste à faire évoluer ce système d'une configuration $x = (x_1, \dots, x_N)$

d'énergie $H(x)$ à une autre configuration $y=(y_1, \dots, y_N)$ d'énergie $H(y)$, jusqu'à atteindre la configuration correspondant à une valeur d'énergie minimale.

Par analogie avec un système physique constitué de particules en interaction, un problème d'optimisation combinatoire peut être résolu, par le recuit simulé, en substituant aux états D_t $t=0..n$ du matériau les configurations x et à l'énergie E la fonction objective H [29].

Il reste que le concept de température pour les systèmes physiques n'a pas de sens dans les problèmes d'optimisation combinatoire. Kirkpatrick et al [31] ont introduit un paramètre contrôlant les passages d'une configuration à une autre et jouant le rôle de la température. Et pour lequel, l'appellation température est maintenue.

1-2-1- Aspects algorithmiques du recuit simulé:

L'algorithme du recuit simulé se présente sous forme d'une séquence d'exécution de l'algorithme de Metropolis [31] [15].

Initialement, on affecte une valeur élevée au paramètre température et on choisit aléatoirement une configuration.

La perturbation aléatoire de l'algorithme de Metropolis est assumée par un générateur de configurations. Le passage d'une configuration x à une configuration y , se fait en choisissant au hasard une configuration y dans le voisinage de x . La probabilité que y soit acceptée comme une nouvelle configuration est:

$$P = \begin{cases} \text{Exp}(-\Delta H/\theta) & \text{si } \Delta H > 0 \\ 1 & \text{sinon} \end{cases} \quad \text{avec } \Delta H = H(y) - H(x) \quad (3.3)$$

La génération des configurations se poursuit jusqu'à l'équilibre où la température est diminuée et la génération reprend la tâche de produire et évaluer les nouvelles configurations. Ce processus est répété jusqu'à la réalisation d'un critère d'arrêt (refroidissement). Ainsi, la configuration finale sera prise comme solution du problème.

La résolution d'un problème d'optimisation combinatoire, par recuit simulé utilise donc, deux mécanismes:

- Le premier consiste à générer les configurations du système et par conséquent les énergies correspondantes.
- Le deuxième appelé schéma de **refroidissement** contrôle la décroissance de la température. Ce schéma permet le passage de la température d'une valeur initiale θ_0 à une valeur minimale θ_{\min} .

- Algorithme du recuit simulé [15]:

Etape d'initialisation:

Entrer la température initiale θ_0 .

Entrer la configuration initiale $x^{(0)} \in \mathcal{X}$, $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)})$.

Calculer $H(x^{(0)})$.

Etape itérative:

Répéter

Répéter

Générer une perturbation Δx , telle que $y = x + \Delta x$

Calculer $\Delta H = H(x + \Delta x) - H(x)$;

Si $\Delta H \leq 0$ **Alors** $x = y$

Sinon Si $\exp(-\Delta H/\theta) \geq \text{random}[0,1]$ **Alors** $x = y$.

Jusqu'à équilibre

Diminuer la température

Jusqu'à condition d'arrêt.

Algorithme 3.1: Recuit simulé.

1-2-2- Aspects mathématiques du recuit simulé:

Des études théoriques sur le recuit simulé considèrent que pour chaque palier de température θ , les configurations constituent une chaîne de Markov [34] [15]. Donc la probabilité de transition $P_{x,y}(\theta)$ d'une configuration x à une configuration y ne dépend que de ces deux configurations. $P_{x,y}(\theta)$ est donnée par:

$$P_{x,y}(\theta) = \begin{cases} \frac{G_{x,y} \cdot \pi_{x,y}(\theta)}{|\mathcal{X}|} & \forall y \neq x \\ 1 - \sum_{z=1, z \neq x} G_{x,z} \cdot \pi_{x,z}(\theta) & \text{si } y = x \end{cases} \quad (3.4)$$

Où $|\mathcal{X}|$: cardinal de \mathcal{X}

$G_{x,y}$: Probabilité d'engendrer la configuration y lorsque la configuration courante est x :

$$G_{x,y} = \begin{cases} \frac{1}{|S_x|} & \text{Si } y \in S_x \\ 0 & \text{Sinon} \end{cases} \quad (3.5)$$

S_x est l'ensemble des configurations obtenues en perturbant x ($S_x \subset \mathcal{X}$).

$\pi_{x,y}(\theta)$: Probabilité d'accepter la configuration y lorsque la configuration courante est x , à la température θ .

1-2-2-1- Paramètres de l'algorithme du recuit simulé:

L'algorithme du recuit simulé est basé sur le calcul de certains paramètres [15]:

- 1/ Valeur initiale de la température θ_0 ,
- 2/ La longueur des chaînes de Markov,
- 3/ Le schéma de décroissance de la température θ ,
- 4/ Condition d'arrêt de l'algorithme.

Plusieurs méthodes de calcul de ces paramètres ont été proposées en littérature:

1/ Valeur initiale de la température θ_0 :

La valeur initiale de la température doit permettre l'acceptation de toutes les configurations visitées ($\forall x, y \in \chi, \pi_{x,y}=1$) [15]. Cette condition permet d'éviter que le système soit piégé par un minimum local.

Kirkpatrick et al [31] proposent de choisir au hasard une très grande valeur pour θ_0 . Si la valeur choisie permet l'acceptation d'au moins 80% des configurations visitées, θ_0 est conservée. Sinon doubler la valeur de θ_0 et recommencer.

Huang et al [27] proposent d'explorer l'espace des configurations χ et de calculer l'écart type σ de la fonction objective. θ_0 est choisie de façon à ce que le passage d'une configuration x à une configuration y est acceptée avec une probabilité P , si $\Delta H = H(y) - H(x) \leq 3\sigma$.

$$\text{D'où} \quad \theta_0 = -\frac{-3\sigma}{\text{Ln } P} \quad (3.6)$$

avec une valeur de P égale à 0.85. [15]

2/ Longueur des chaînes de Markov:

Cette valeur notée L_m correspond au nombre d'itérations, dans un même palier de température (θ constante). Cette valeur peut être déterminée par le nombre minimal de configurations acceptées dans une même chaîne [15]. Lorsque le nombre de configurations acceptées dans une même chaîne atteint un seuil, le paramètre de température est diminué.

Kirkpatrick et al [30], Aart et Larhoven [2] et John [28] proposent de borner L_m par $|S_x|$ pour $x \in \chi$ et toutes les chaînes ont alors la même longueur.

3/ Le schéma de décroissance de la température θ :

La décroissance de la température θ , doit se faire lentement. Ce mécanisme détermine le nombre de chaînes de Markov (nombre de paliers de température) durant l'exécution du recuit simulé.

Kirkpatrick et al [31][15] proposent la règle de décroissance suivante :

$$\theta_{t+1} = \alpha \cdot \theta_t \quad \text{avec } \alpha=0.95 \quad (3.7)$$

Huang et al [27][15] proposent une autre règle basée sur l'écart type de la fonction objective pour deux chaînes de Markov consécutives:

$$\theta_{t+1} = \theta_t \cdot \text{Exp}\left(\frac{-\lambda \theta_t}{\sigma(\theta_t)}\right) \quad (3.8)$$

Où

$\sigma(\theta_t)$: écart type de la fonction objective des configurations du palier de température θ_t .

$$0 < \lambda \leq 1$$

$\frac{\theta_{t+1}}{\theta_t} \geq 0.6$. Cette condition permet d'éviter une décroissance brusque de la température.

4/ Condition d'arrêt de l'algorithme:

Selon Kirkpatrick et al [31] et Sechen et al [47], l'algorithme doit s'arrêter lorsque les configurations engendrées par des chaînes de Markov consécutives restent inchangées pour un certain nombre de chaînes.

Bonomi et Lutton [10] proposent de fixer ce nombre à une valeur comprise entre 30 et 50 selon le problème à traiter.

Huang et al [27] proposent de comparer à la fin de chaque chaîne de Markov, l'écart entre les valeurs minimale et maximale de la fonction objective, avec la plus grande variation de H parmi les configurations acceptées.

Si les valeurs comparées sont égales, toutes les configurations de la chaîne courante ont le même coût, il est donc inutile de continuer le recuit simulé [15].

Xu et Erkki [55][18] proposent de fixer un seuil minimal pour la température, pour que l'algorithme s'arrête.

1-2-3- Utilisation du recuit simulé:

Le recuit simulé est utilisé dans les problèmes d'optimisation combinatoire (voyageur de commerce [9][16], problème d'ordonnancement [18]...) et en traitement d'images notamment en restauration [54][56].

2- LE RECUIT SIMULE EN SEGMENTATION DES IMAGES:

Rappelons que la segmentation des images vise à réduire le nombre considérable d'informations contenues dans l'image brute (initiale), en regroupant les pixels ayant une caractéristique commune.

Nous nous proposons dans ce contexte d'utiliser l'approche du recuit simulé, dans le but de réaliser l'opération de segmentation d'image à niveau de gris. Ainsi, nous présentons trois algorithmes, basés sur le recuit simulé et caractérisés chacun par le problème à optimiser, où sont définis :

- L'espace Ω des sites.
- L'espace χ des configurations.
- La fonction objective H à minimiser.
- Le schéma de génération des configurations.
- Les 4 paramètres du recuit présentés en (§ 1.2.2.1 p. 67).

2-1- Modèle de segmentation <Recuit 1>:

2-1-1- Principe de la méthode:

Rappelons que les images satellitaires, même si elles présentent un histogramme de niveau de gris unimodal, elles sont composées de plusieurs objets dont les histogrammes des niveaux de gris respectifs se recouvrent (Figure 3.3- (a)).

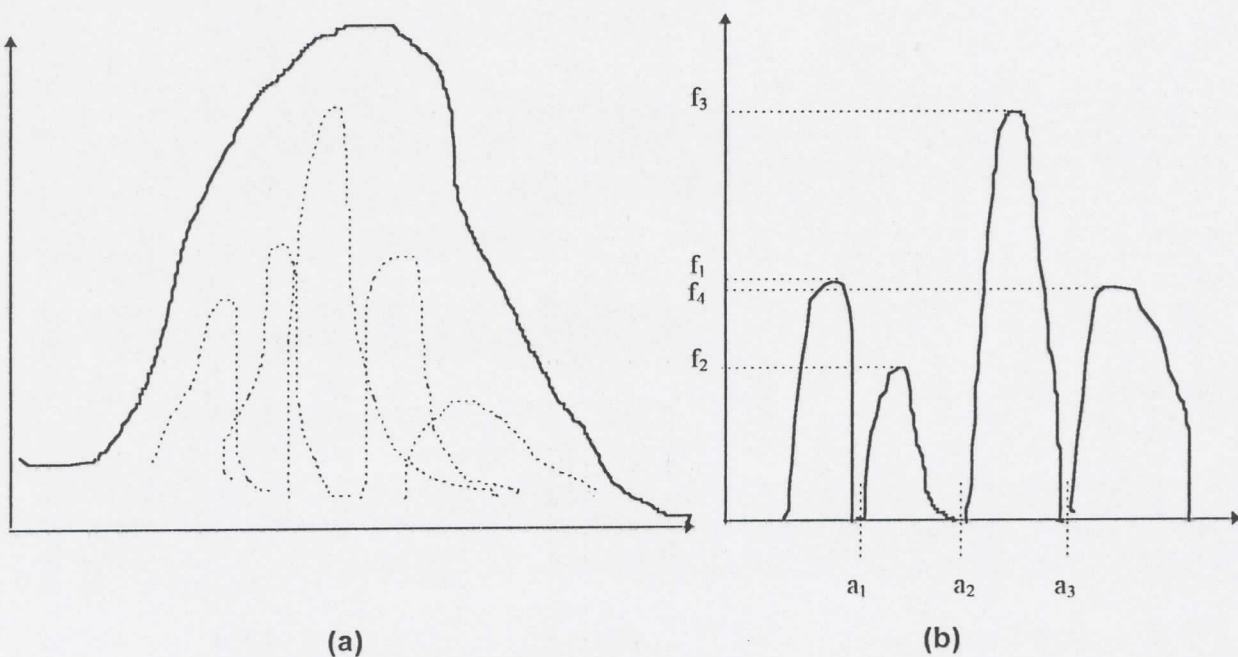


Figure 3.3 : (a) Contribution des histogrammes de niveaux de gris de plusieurs objets à l'historgramme global de l'image.
(b) Fragmentation de l'historgramme global.

Ce modèle a pour but la répartition des pixels de l'image dans des classes de niveaux de gris déterminés. Il est basé sur la fragmentation de l'histogramme global de l'image désigné par $Hist$, en des sous-histogrammes $Hist_m$ $m=1..M$, représentés chacun par une fréquence maximale f_m et sont délimités deux à deux par un niveau de gris seuil a_m $m=1..M-1$ (Figure 3.3 (b)).

L'algorithme procède par isolation progressive des histogrammes, selon un recuit simulé. A chaque étape t , ces différents histogrammes sont représentés par leurs fréquences maximales $f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)}$ et par les seuils les délimitant $a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)}$. La perturbation des paramètres $(f_1, f_2, \dots, f_M; a_1, a_2, \dots, a_{M-1})$, qui constituent en fait une configuration de notre système, permet d'éloigner les histogrammes l'un de l'autre, en concentrant aux mieux les points qui les composent, autour d'une valeur de gris moyenne.

Enfin, à partir de la configuration retenue comme solution du système et suivant le principe du seuillage, on procède à l'affectation d'un pixel à une classe en comparant son niveau de gris aux seuils calculés.

2-1-2- Aspects théoriques de l'algorithme:

2-1-2-1- Modélisation:

Soient: $G(L,K)$: la matrice des niveaux de gris de l'image initiale.

E : échelle des niveaux de gris: $E=[0,255] \cap \mathbf{N}$. ($\mathbf{N} = \{ \text{entiers naturels} \}$)

$Hist$: vecteur histogramme des niveaux de gris de G .

$Hist_1, Hist_2, \dots, Hist_M$: histogrammes formant l'histogramme global de l'image.

f_1, f_2, \dots, f_M : les fréquences maximales respectives de $Hist_1, Hist_2, \dots, Hist_M$

$$\forall m=1..M, f_m \in F = [0, L \times K] \cap \mathbf{N}$$

a_1, a_2, \dots, a_{M-1} niveaux de gris séparant deux classes consécutives.

$$\forall m=1..M-1, a_m \in E$$

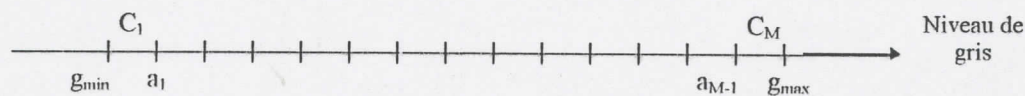
$x = (f_1, f_2, \dots, f_M; a_1, a_2, \dots, a_{M-1})$ une configuration du système.

χ : espace des configurations possibles: $\chi = F^M \times E^{M-1}$

2-1-2-2- Paramètres de Recuit1:

1/ Génération de la configuration initiale:

Initialement, on détermine g_{\min} et g_{\max} les niveaux de gris respectivement minimal et maximal de l'image initiale G . L'intervalle $[g_{\min}, g_{\max}]$ est partagé en M sous intervalles (classes).



Les classes C_m , $m=1\dots M$, sont définies respectivement par les bornes $[g_{\min}, a_1]$, $[a_1+1, a_2], \dots, [a_{M-1}+1, g_{\max}]$ et par les fréquences maximales f_1, f_2, \dots, f_M . Ces paramètres constituent la configuration initiale du système $x^{(0)} = (f_1^{(0)}, f_2^{(0)}, \dots, f_M^{(0)}; a_1^{(0)}, a_2^{(0)}, \dots, a_{M-1}^{(0)})$ et sont choisis aléatoirement, avec les conditions suivantes:

- $g_{\min} < a_1^{(0)} < a_2^{(0)} < \dots < a_{M-1}^{(0)} < g_{\max}$
- $0 < f_1^{(0)} + f_2^{(0)} + \dots + f_M^{(0)} \leq L \times K$
- $f_m^{(0)} \leq \text{Entier}(K \times L / M)$

La première condition sur les niveaux de gris représentatifs des différentes classes, permet d'éviter un chevauchement entre celles-ci. La seconde maintient l'effectif de l'image à $K \times L$ pixels. La troisième condition, permet la normalisation des fréquences en fonction du nombre de classes.

2/ Fonction objective à minimiser:

$$H_t = H(x^{(t)}) = \sum_{i=g_{\min}}^{a_1^{(t)}} |\text{Hist}[i] - f_1^{(t)}| + \sum_{i=a_1^{(t)}+1}^{a_2^{(t)}} |\text{Hist}[i] - f_2^{(t)}| + \dots + \sum_{i=a_{M-1}^{(t)}+1}^{g_{\max}} |\text{Hist}[i] - f_M^{(t)}| \quad (3.9)$$

La fonction H est égale à la somme absolue des écarts entre les valeurs de l'histogramme initial et les valeurs des fréquences calculées à l'itération t , pour les différentes classes.

La minimisation de la fonction H permet donc, de réduire l'écart entre l'histogramme initial et l'histogramme obtenu à l'itération t .

La procédure **Hist-Energie(H_t)** permet l'évaluation de la fonction Energie de (3.9):

```

Début
   $H_t \leftarrow 0$ ;  $a_0 \leftarrow g_{\min} - 1$ ;  $a_M \leftarrow g_{\max}$ ;
  Pour  $m \leftarrow 1$  à  $M$ 
  Faire
    Pour  $i \leftarrow a_{m-1} + 1$  à  $a_m$ 
    Faire
       $H_t \leftarrow H_t + \text{abs}(\text{Hist}[i] - f_m^{(t)})$ 
    Fait;
  Fait;
Fin

```

Algorithme 3.2 : Procédure Hist_Energie (H_t).

3/ Schéma de perturbation et génération des configurations:

Une configuration $x^{(t+1)}$ est générée en perturbant la configuration courante $x^{(t)} = (f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)}; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)})$.

- Perturbation des fréquences courantes $f_m^{(t)}$ et génération de $f_m^{(t+1)}$, $m=1..M$:

$$f_m^{(t+1)} = f_m^{(t)} \pm f$$

f est une petite valeur entière. f doit vérifier la condition:

$$0 < f_1^{(t+1)} + f_2^{(t+1)} + \dots + f_M^{(t+1)} \leq L \times K$$

- Perturbation des seuils courants $a_m^{(t)}$ et génération de $a_m^{(t+1)}$, $m=1..M-1$:

$$a_m^{(t+1)} = a_m^{(t)} \pm \delta$$

δ est une faible variation de niveau de gris ($1 \leq \delta \leq 3$) et vérifie les conditions:

- $a_m^{(t+1)} \in E$
- $g_{\min} < a_1^{(t+1)} < a_2^{(t+1)} < \dots < a_{M-1}^{(t+1)} < g_{\max}$

La procédure **Hist_Generer**($f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)}; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)}$) permet la perturbation de la configuration courante et la génération d'une nouvelle configuration:

Début

(Génération des fréquences f_m , $m=1..M$)

Répéter

$m \leftarrow 1$;

Tant que $m \leq M$

Faire Générer aléatoirement une valeur entière :

$$f \leftarrow \text{random}(L \times K / M)$$

Générer une probabilité :

$$\varepsilon \leftarrow \text{random}(101) / 100$$

Si ($\varepsilon \leq 1/2$ et $(f_m^{(t)} - f) \leq K \times L$) alors $f_m^{(t+1)} \leftarrow f_m^{(t)} - f$

Si ($\varepsilon > 1/2$ et $(f_m^{(t)} + f) \leq K \times L$) alors $f_m^{(t+1)} \leftarrow f_m^{(t)} + f$

Fait;

Jusqu'à $\sum_{m=1}^M f_m^{(t+1)} \leftarrow K \times L$

(Génération des seuils a_m , $m=1..M-1$)

$m \leftarrow 1$;

Tant que $m \leq M-1$

Faire Générer aléatoirement une faible variation de niveau de gris:

$$\delta \leftarrow \text{random}(3);$$

Générer une probabilité:

$$\varepsilon \leftarrow \text{random}(101) / 100$$

Si ($\varepsilon \leq 1/2$ et $(a_m^{(t)} - \delta) \in E$) alors $a_m^{(t+1)} \leftarrow a_m^{(t)} - \delta$

Si ($\varepsilon > 1/2$ et $(a_m^{(t)} + \delta) \in E$) alors $a_m^{(t+1)} \leftarrow a_m^{(t)} + \delta$

Si $a_m^{(t+1)} > a_{m-1}^{(t+1)}$ alors $m \leftarrow m+1$;

Fait;

Fin

Algorithme 3.3 : Hist_generer ($f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)}; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)}$).

4/ Valeur initiale de la température:

On choisit au hasard une très grande valeur de la température.

5/ Longueur des chaînes de Markov:

Les configurations obtenues dans un même palier de température constituent une chaîne de Markov. La longueur de cette chaîne est fonction de H . Si cette dernière reste constante ($\Delta H=0$), on diminue la température et on passe à un nouveau palier. C'est le principe utilisé par Xu et Erkki dans [55][18].

6/ Règle de décroissance de la température (schéma de refroidissement):

Le schéma de refroidissement utilisé dans ce modèle est celui proposé par Kirpatrick [31]:

$$\theta_{t+1} = \alpha \cdot \theta_t \quad \text{avec } \alpha=0.95$$

La décroissance de θ est fonction du nombre d'itérations. La décroissance est d'autant plus lente que le nombre d'itérations augmente.

7/ Condition d'arrêt de l'algorithme :

La condition d'arrêt que nous avons utilisé est celle proposée par Xu [55]; l'exécution du recuit est interrompue si la température atteint un seuil minimal.

2-1-2-3- Opération de seuillage :

Après avoir déterminé la solution optimale $x^*=(f_1, f_2, \dots, f_M; a_1, a_2, \dots, a_{M-1})$, on procède à l'opération de seuillage: On attribue à chaque pixel (i, j) une nouvelle valeur de niveau de gris $G_s[i, j]$, en fonction de son niveau de gris initial $G[i, j]$ et en fonction des seuils déterminés $(a_1, a_2, \dots, a_{M-1})$.

<p>Début</p> <p>Pour $i \leftarrow 1$ à K</p> <p>Faire Pour $j \leftarrow 1$ à L</p> <p>Faire Si $G[i, j] \leq a_1$ alors $G_s[i, j] \leftarrow \text{entier} (a_1 + g_{\min}) / 2$ fsi;</p> <p>Si $(a_1 + 1 \leq G[i, j] \leq a_2)$ alors $G_s[i, j] \leftarrow \text{entier} (a_1 + 1 + a_2) / 2$ fsi;</p> <p>.....</p> <p>Si $a_{M-1} + 1 \leq G[i, j]$ alors $G_s[i, j] \leftarrow \text{entier} (a_{M-1} + 1 + g_{\max}) / 2$ fsi;</p> <p>Fait;</p> <p>Fait;</p> <p>Fin;</p>
--

Algorithme 3.4: Procédure Seuillage.

2-1-2-4- Algorithme: Recuit 1

```

Début
  Lecture de la température initiale  $\theta_0$  et de la température seuil  $\theta_{\min}$ .
  Lecture de l'image initiale:  $G(L,K)$ 
  Calcul de l'histogramme HIST de l'image  $G$ .
   $t \leftarrow 0$  , Choisir aléatoirement :  $x^{(0)} = (f_1^{(0)}, f_2^{(0)}, \dots, f_M^{(0)} ; a_1^{(0)}, a_2^{(0)}, \dots, a_{M-1}^{(0)})$ 
  Hist_energie( $H_0$ );
1-  $t \leftarrow t+1$ 
  Hist_generer ( $f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)} ; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)}$ );
  Hist_energie( $H_t$ );
   $\Delta H \leftarrow H_t - H_{t-1}$ 
  Si  $\Delta H < 0$  alors accepter  $x_1^{(t)} = (f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)} ; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)})$ ; aller à 1; fsi;
  Si  $\Delta H > 0$  alors accepter  $x_1^{(t)} = (f_1^{(t)}, f_2^{(t)}, \dots, f_M^{(t)} ; a_1^{(t)}, a_2^{(t)}, \dots, a_{M-1}^{(t)})$  avec une probabilité
     $\exp(-\Delta H / \theta_t)$  ; aller à 1 fsi;
  Si  $\Delta H = 0$  alors  $\theta_t \leftarrow 0.95 * \theta_{t-1}$ 
    Si  $\theta_t < \theta_{\min}$  alors Fin
    sinon aller à 1
  fsi;
fsi;
Fin;

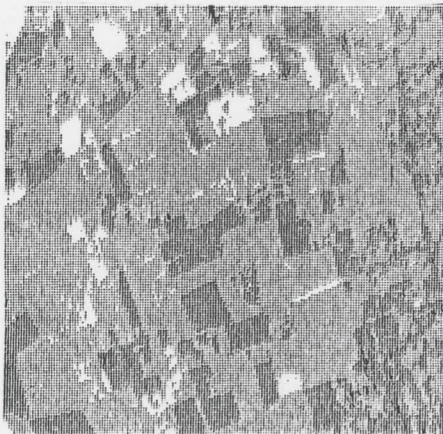
```

Algorithme 3.5: Recuit1.

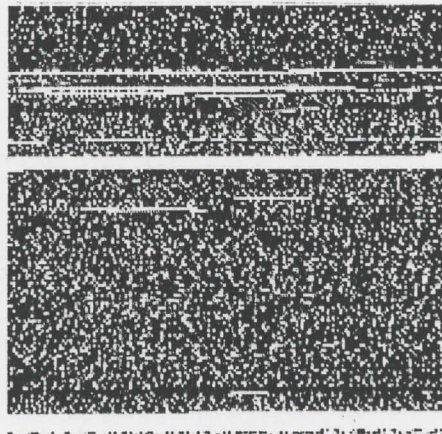
2-1-3- Application de Recuit1:

Données: L'algorithme Recuit1 est appliqué sur une fenêtre de l'image SPOT XS1, de résolution spatiale 20x20m et représentant la région de l'Atlas Blidéen. En faisant varier les paramètres température initiale , la température minimale. Le nombre de classes M étant fixé à 3.

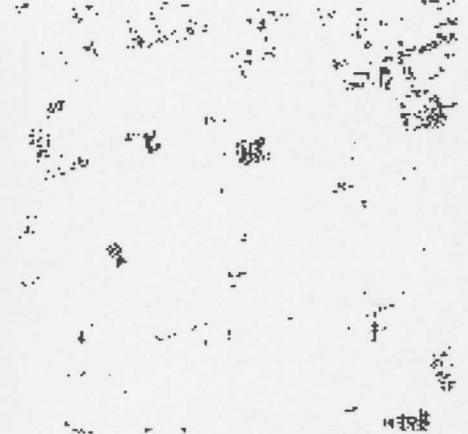
2-1-3-1- Affichage des résultats:



1-) Image originale
170 x 170



2-) Image résultat
Cas a : $\theta_0=100$ $\theta_{\min}=1$



3-) Image résultat
Cas b: $\theta_0=10$ $\theta_{\min}=0.01$

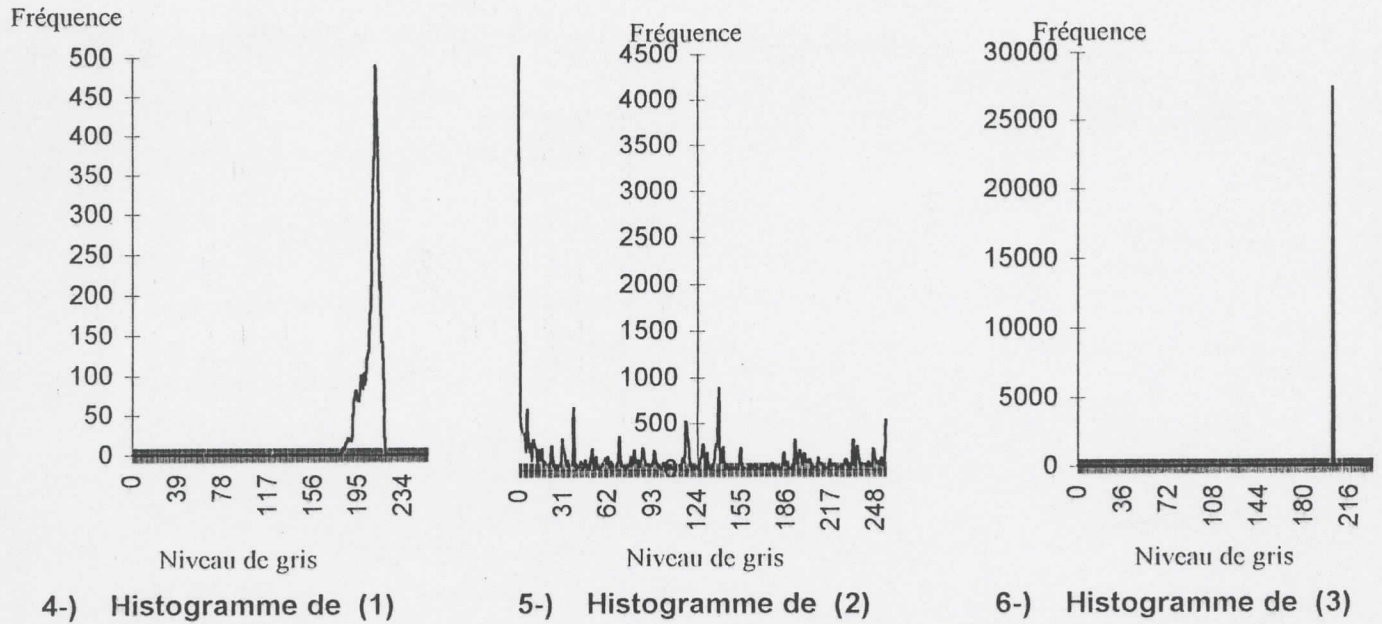
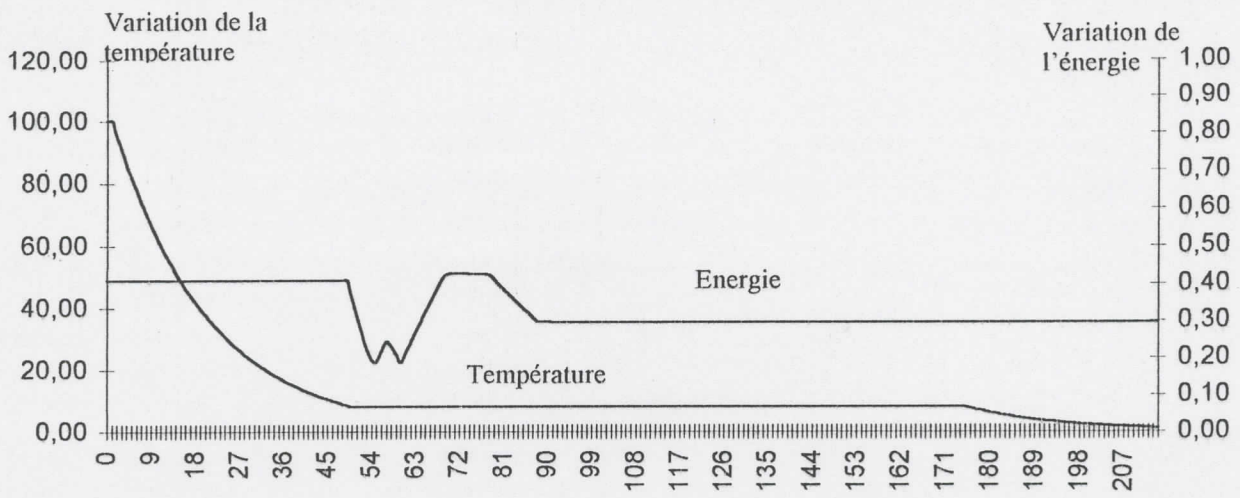
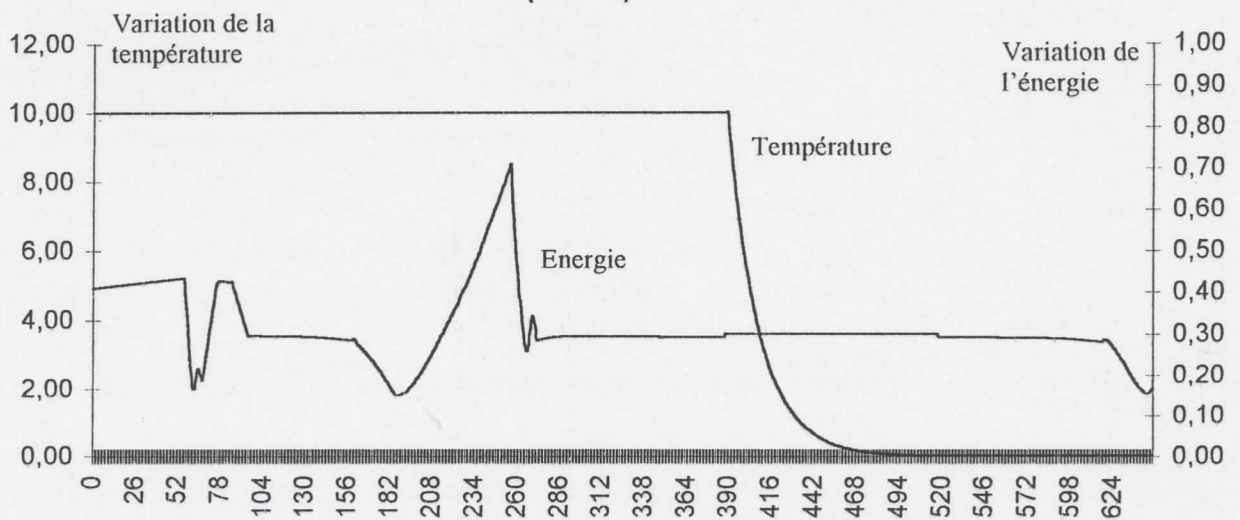


Figure 3.4: Application de Recuit1 : Images et Histogrammes de Niveau de gris.



(Cas a)



(Cas b)

Figure 3.5: Application de Recuit1. Variation de l'énergie et de la température.

2-1-3-2- Interprétation des résultats de l'algorithme Recuit1:

Cas	Nbre de classes M	O_0	O_{\min}	Solution finale x^*					Nombre d'itérations	Température finale	Energie finale
				a_1	a_2	f_1	f_2	f_3			
a-)	3	100	1	11	20	2229	451	0	215	0.99	0.29
b-)	3	10	0.01	26	95	212	27343	41	650	0.009	0.16

Tableau 3.1: Résultats de l'application de Recuit1.

L'algorithme Recuit1 vise à transformer l'histogramme de l'image originale, en M classes, délimitées par des niveaux de gris déterminés. Cette transformation se fait progressivement, par recuit simulé où on cherche le minimum de la fonction énergie définie en (3.9). Après avoir déterminé la solution optimale présumée $x^* = (f_1, f_2, \dots, f_M; a_1, a_2, \dots, a_{M-1})$ de ce problème, on procède à l'affectation des pixels dans les classes selon la procédure seuillage (Algorithme 3.4).

Parmi les différentes applications de Recuit1, nous avons présenté deux cas (a) et (b): où on fait varier les paramètres température initiale et température minimale, le nombre de classes étant égal à 3.

CAS a: L'observation de la figure (3.4 image2), nous permet de constater que cette image ne représente pas le résultat attendu en segmentation. Son histogramme présente un pic de mode zéro et de fréquence 4200, ainsi que plusieurs autres pics de fréquences variables. La courbe de variation de la fonction énergie et de la température (figure 3.5 cas a), montre que la fonction énergie atteint son minimum à l'itération 60, à une température relativement élevée. Ceci a permis de quitter ce puits d'énergie à la recherche d'un autre minimum qui malheureusement n'a pu être déterminé, vu que l'énergie s'est stabilisée à une valeur constante jusqu'à l'arrêt de l'algorithme.

CAS b: L'image de la figure (3.4 image3), présente un meilleur résultat en segmentation relativement au cas (a). On voit l'apparition de quelques zones homogènes, où les pixels ont été classés correctement. Son histogramme présente un pic de mode 206 et de fréquence 27343, ainsi que deux autres pics respectivement de fréquences 212 et 41 et de modes 25 et 115. La courbe de variation de la fonction énergie et de la température (figure 3.5 cas b), montre qu'au cours de l'exécution de l'algorithme la fonction énergie atteint des minimums locaux. Au bout de 650 itérations, l'algorithme s'arrête et la fonction énergie atteint un autre minimum qui correspond à la solution retenue.

La comparaison des résultats obtenus dans les deux cas, montre que le résultat du recuit simulé est d'autant meilleur que la décroissance de la température est lente.

2-2- Modèle de segmentation: <Recuit 2>

2-2-1- Principe de la méthode:

L'algorithme Recuit2 procède par répartition des pixels (s_1, \dots, s_N) de l'image selon leurs niveaux de gris initiaux (d_1, \dots, d_N) , en M classes C_m , $m=1, \dots, M$, caractérisées chacune par un niveau de gris déterminé a_m , $m=1, \dots, M$.

L'affectation des pixels aux classes C_m , se fait d'une manière progressive par recuit simulé. A chaque étape on perturbe les niveaux de gris a_m caractérisant les classes C_m , $m=1, \dots, M$ et on propose une nouvelle configuration $x=(x_1, \dots, x_N)$ où $x_i \in \{a_1, \dots, a_m\}$. La configuration retenue comme solution du problème serait la configuration présentant un écart quadratique minimal avec l'image initiale.

2-2-2- Aspects théoriques de l'algorithme:

2-2-2-1- Modélisation:

Soient: $A(L,K)$: la matrice image, on l'assimile à Ω , un espace fini de sites (pixels) de taille $N=L \times K$.

Un pixel $A[i,j]$ de l'image est noté s , un site de l'espace Ω .

E : espace des états possibles d'un pixel (échelle des niveaux de gris):

$$E = [0, 255] \cap \mathbf{N}. \quad (\mathbf{N} = \{\text{entiers naturels}\})$$

χ : espace des configurations possibles ($\chi \subseteq E^N$)

d : image initiale à segmenter (donnée), telle que :

$$\begin{aligned} d: \Omega^N &\longrightarrow \chi \\ (s_1, \dots, s_N) &\longrightarrow d(s_1, \dots, s_N) = (d_1, \dots, d_N) \end{aligned}$$

Où d_i niveau de gris initial du pixel s_i , $i=1..N$.

x : configuration image proposée :

$$\begin{aligned} x: \Omega^N &\longrightarrow \chi \\ (s_1, \dots, s_N) &\longrightarrow x = (x_1, \dots, x_N) \end{aligned}$$

Où x_i niveau de gris proposé du pixel s_i , $i=1..N$.

2-2-2-2- Paramètres de Recuit2:

1/ Génération de la configuration initiale:

On choisit aléatoirement M niveaux de gris $a_1^{(0)}, a_2^{(0)}, \dots, a_M^{(0)}$, représentant M classes. Les valeurs de la configuration initiale $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)})$ sont choisies aléatoirement dans l'ensemble $\{a_1^{(0)}, a_2^{(0)}, \dots, a_M^{(0)}\}$. (ie $\forall i=1..N, x_i^{(0)} \in \{a_1^{(0)}, a_2^{(0)}, \dots, a_M^{(0)}\}$).

2/ Fonction objective à minimiser:

$$H_t = \sum_{i=1}^N (d_i - x_i^{(t)})^2 \quad (3.10)$$

La fonction H est égale à la somme des écarts quadratiques entre les valeurs de gris initiaux des pixels et les valeurs de gris proposés à l'itération t.

La minimisation de la fonction H permet donc, de réduire l'écart entre l'image initiale d et l'image proposée x. Cela assure une bonne correspondance entre l'image d et la solution du problème retenue.

La procédure **Img_Energie(H_t)** permet l'évaluation de la fonction objective H_t:

```

Début
  Ht ← 0;
  Pour i ← 1 à N
  Faire
    Ht ← Ht + (di - xi(t))2
  Fait;
Fin

```

Algorithme 3.6 : **Img_Energie(H_t)**.

3/ Schéma de perturbation et génération des configurations:

- Perturbation des niveaux de gris a_m, m=1..M:

$$a_m^{(t+1)} = a_m^{(t)} \pm \delta$$

δ est une faible variation de niveau de gris (1 ≤ δ ≤ 3) et doit vérifier la condition « a_m^(t+1) ∈ E ».

La procédure **Img_perturber(a₁^(t), a₂^(t), ..., a_M^(t))** permet la perturbation des niveaux de gris :

```

Début
  m ← 1;
  Tant que m ≤ M
  Faire Générer aléatoirement une faible variation de niveau
    de gris : δ ← random(3)+1
    Générer une probabilité :
      ε ← random[101]/100
    Si (ε ≤ ½ et (am(t) - δ) ∈ E) alors am(t+1) ← am(t) - δ
    Si (ε > ½ et (am(t) + δ) ∈ E) sinon am(t+1) ← am(t) + δ
  Fait;
Fin

```

Algorithme 3.7 : **Img_perturber(a₁^(t), a₂^(t), ..., a_M^(t))**.

- Génération des configurations:

Une nouvelle configuration $x^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_N^{(t+1)})$ est choisie aléatoirement parmi les valeurs de $\{a_1^{(t+1)}, a_2^{(t+1)}, \dots, a_M^{(t+1)}\}$.

La procédure **Img_generer** ($x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}$) permet la génération des configurations x :

```

Début
  Pour i ← 1 to N
    Faire m ← random(M)+1;
            $x_i^{(t+1)} \leftarrow a_m^{(t+1)}$ ;
    Fait;
  Fin

```

Algorithme 3.8 : **Img_generer ($x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}$).**

4/ Valeur initiale de la température:

On choisit au hasard une très grande valeur de la température.

5/ Longueur des chaînes de Markov:

Les configurations obtenues dans un même palier de température constituent une chaîne de Markov. La longueur de cette chaîne est fonction de H . Si cette dernière reste constante ($\Delta H=0$), on diminue la température et on passe à un nouveau palier. C'est le principe utilisé par Xu et Erkki dans [55].

6/ Règle de décroissance de la température (schéma de refroidissement):

Afin d'assurer la décroissance lente de la température, nous utilisons une constante positive inférieure à 1, qui réalise le refroidissement du système en fonction du nombre d'itérations [31].

$$\theta_{t+1} = 0.95 \cdot \theta_t$$

7/ Condition d'arrêt de l'algorithme :

La condition d'arrêt que nous avons utilisé est celle proposée par Xu [55]; l'exécution du recuit est interrompue si la température atteint un seuil minimal choisi.

2-2-2-3- Algorithme: Recuit2**Début**

Lecture de la température initiale θ_0 et de la température seuil θ_{\min} .

Lecture de l'image initiale: $d(s_1, \dots, s_N) = (d_1, \dots, d_N)$;

$t \leftarrow 0$, Choisir aléatoirement : $(a_1^{(0)}, a_2^{(0)}, \dots, a_M^{(0)})$;

Img_generer $(x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)})$;

Img_energie(H_0);

1- $t \leftarrow t+1$

Img_perturber($a_1^{(t)}, a_2^{(t)}, \dots, a_M^{(t)}$);

Img_generer($x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)}$);

Img_energie(H_t);

$\Delta H \leftarrow H_t - H_{t-1}$

Si $\Delta H < 0$ **alors** accepter $(x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)})$; aller à 1; **fsi**;

Si $\Delta H > 0$ **alors** accepter $(x_1^{(t)}, x_2^{(t)}, \dots, x_N^{(t)})$ avec une probabilité $\exp(-\Delta H / \theta_t)$; aller à 1 **fsi**;

Si $\Delta H = 0$ **alors** $\theta_t \leftarrow 0.95 * \theta_{t-1}$

Si $\theta_t < \theta_{\min}$ **alors** Fin

sinon aller à 1

fsi;

fsi;

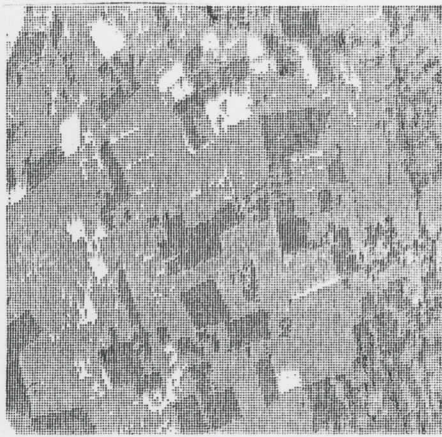
Fin;

Algorithme 3.9: Recuit2.

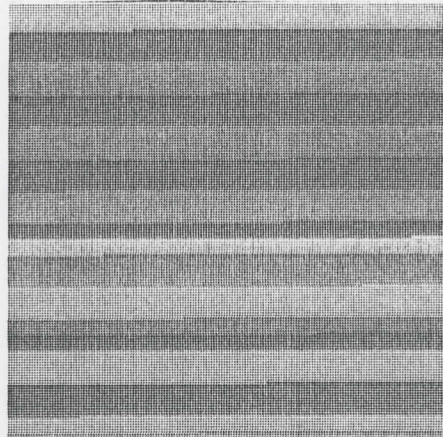
2-2-3- Application de Recuit2:

Données: L'algorithme Recuit2 est appliqué sur des fenêtres de l'image SPOT XS1, de résolution spatiale 20x20m et représentant la région de l'Atlas Blidéen. En faisant varier les paramètres température initiale, la température minimale, pour un nombre de classes égal à 3.

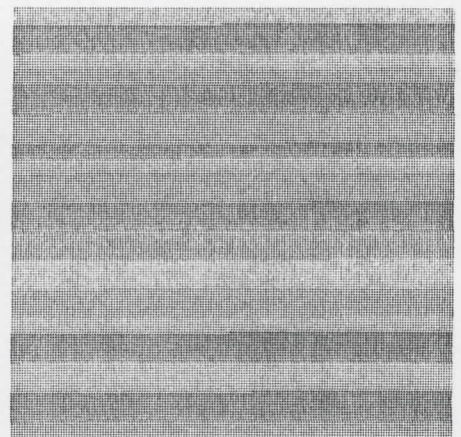
2-2-3-1- Affichage des résultats:



1-) Image originale
170 x 170

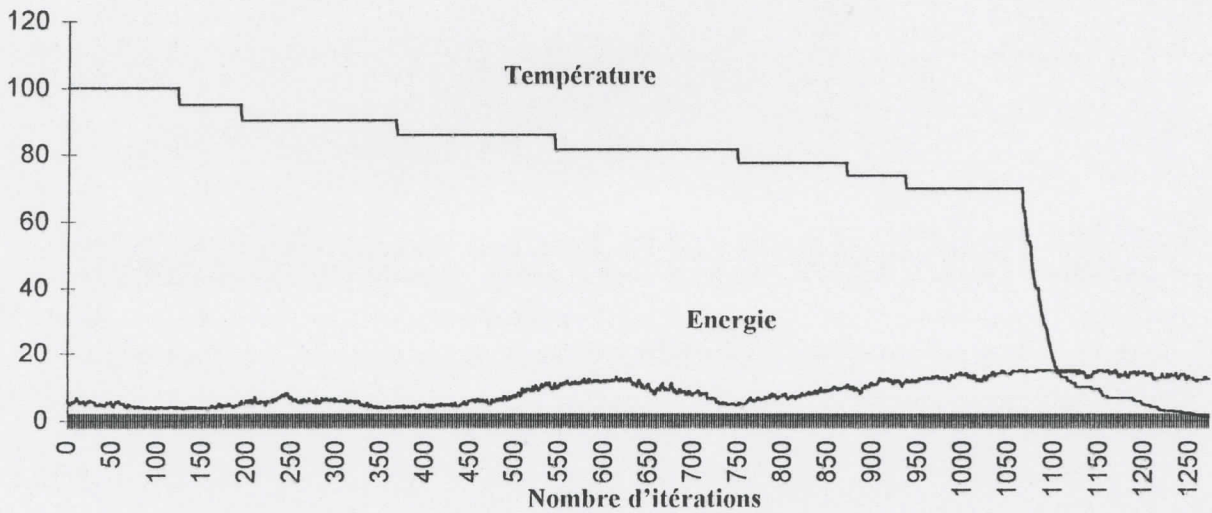


2-) Image résultat
Cas a: $\theta_0 = 100$ $\theta_{\min} = 1$

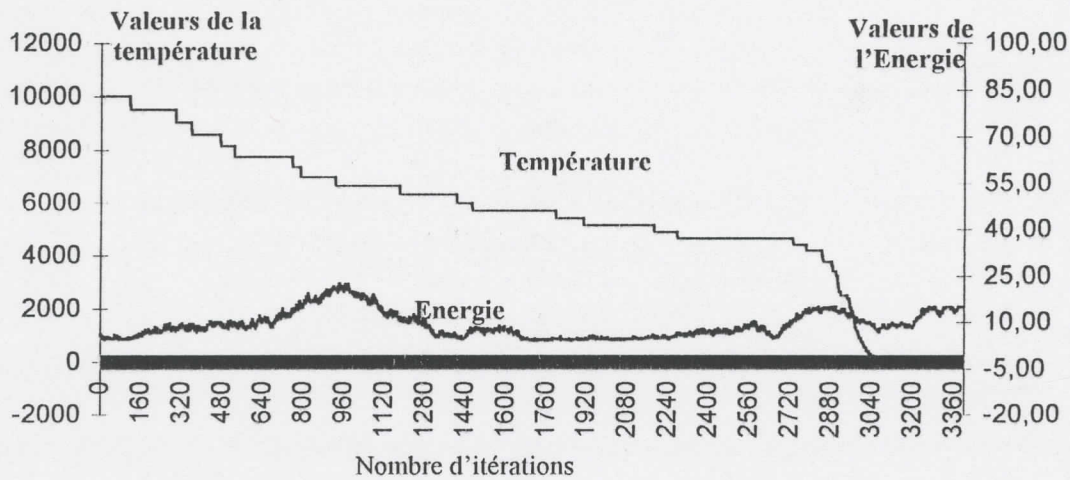


3-) Image résultat
Cas b: $\theta_0 = 10\ 000$ $\theta_{\min} = 0.01$

Figure 3.6: Images obtenues par application de Recuit2.



Cas (a)



Cas (b)

Figure 3.7: Application de Recuit2: Variation de l'énergie et de la température. Cas: a et b.

2-2-3-2- Interprétation des résultats de l'algorithme Recuit2:

Cas	θ_0	θ_{\min}	Niveaux de gris a_1, \dots, a_M	Nombre d'itérations	Température finale	Energie finale
a-)	100	1	160 175 195	1273	0.988	12.36
b-)	10 000	0.01	202 248 249	3400	0.009	14.6

Tableau 3.2: Résultats de l'application de Recuit2.

L'algorithme Recuit2 vise à affecter les pixels de l'image dans des classes de niveau de gris déterminé. Partant d'une configuration initiale aléatoire de l'image, l'algorithme affecte les pixels dans les différentes classes, d'une manière progressive, par recuit simulé, en minimisant à chaque étape, l'écart quadratique entre l'image initiale et la nouvelle configuration proposée.

Dans l'application de Recuit2, on fait varier les paramètres température initiale et température minimale, pour un nombre de classes égal à 3.

CAS a: On fixe l'intervalle de température à $[100,1]$, l'image obtenue (Figure 3.6 image2), se présente sous formes de bandes de 3 classes de niveaux de gris, ne correspondant pas à la disposition géométrique des différentes zones de l'image initiale. La courbe de variation de la fonction énergie et de la température (Figure 3.7 cas a), montre que la fonction énergie rencontre plusieurs minimum locaux, ne correspondant pas à la solution optimale.

CAS b: Dans ce cas on étend l'intervalle de température, afin de permettre un plus grand nombre d'itérations que le cas précédent. Le résultat reste non satisfaisant, l'image se présente sous forme de bandes de 3 niveaux de gris (Figure 3.6 image 3). La courbe de variation de la fonction énergie (Figure 3.7 cas b), se présente sous forme d'oscillations, la fonction énergie rencontre un minimum local (puits d'énergie), puis augmente et quitte le puits, rencontre un autre minimum local, et ainsi de suite jusqu'à ce que l'algorithme s'arrête, sans que la fonction énergie atteigne son minimum optimal.

A partir de ces résultats, on conclut que l'algorithme Recuit 2, n'a pas donné des résultats satisfaisants en segmentation d'images.

2-3- Recuit Simulé avec Champs de Markov (Recuit3):

Dans l'algorithme précédent Recuit 2, l'image est traitée globalement sans tenir compte de l'interaction qui existe entre les pixels voisins. Or, l'image du monde réel possède une certaine redondance spatiale: des pixels voisins ont généralement des caractéristiques communes[52]. Afin de mettre en évidence cet aspect, nous adoptons pour ce troisième algorithme, l'utilisation des champs de Markov en segmentation d'images.

Les champs de Markov ont fait une apparition remarquable dans le domaine de traitement d'images, au début des années 1980 [35]. Ils permettent d'introduire des contraintes liées à la notion de système de voisinage qui constitue une notion de base en imagerie [14].

2-3-1- Généralités sur les champs de Markov:

2-3-1-1- Définition d'un champ aléatoire[14][54]:

Soit Ω un ensemble fini de N sites, représentant la matrice image. Les sites étant les pixels et soit $E=[0,255] \cap \mathbf{N}$. ($\mathbf{N}=\{\text{entiers naturels}\}$) ensemble de 256 niveaux de gris.

A chaque site s de Ω , on associe une variable aléatoire X_s , à valeurs dans E . L'image est donc considérée comme un vecteur aléatoire $X = (X_s, s \in \Omega)$. La réalisation $x = (x_s, s \in \Omega)$ de X est une configuration de l'espace $\chi \subseteq E^N$. Le vecteur aléatoire X est un *champ aléatoire*.

On considère les probabilités suivantes:

$P(X_s = x_s)$: Loi marginale associée au site s .

$P(X = x) = P(x)$: Probabilité de la configuration x , avec $P(x) \geq 0$ et $\sum_{x \in X} P(x) = 1$

La distribution $P(x)$, strictement positive, est aussi appelée *champ aléatoire* ou *champ stochastique*.

2-3-1-2- Systèmes de voisinage et cliques [35][14]:

• Un système de voisinage sur Ω est un ensemble $V = \{V_s \subset \Omega; s \in \Omega\}$ des parties de Ω tel que:

- 1) $s \notin V_s$
- 2) $\forall (s, r)^2 \in \Omega^2, s \in V_r \Leftrightarrow r \in V_s.$

• Un sous-ensemble C de Ω est appelé **clique** relativement au système de voisinage V si C vérifie:

- 1) C est un singleton: $|C|=1$.
- Ou
- 2) $|C| > 1$ et deux éléments quelconques de C sont toujours voisins.

Le cardinal de C , $|C|$ est appelé ordre de la clique C .

Dans le cas d'un système de voisinage à 4 (chaque site s a exactement quatre voisins), il y a 2 types de cliques: les singletons et 4 cliques d'ordre 2.

	S ₁	
S ₂	s	S ₃
	S ₄	

Cliques du 4-voisinage :

- Les singletons.
- Cliques d'ordre 2: $\{s, s_1\}$; $\{s, s_2\}$; $\{s, s_3\}$; $\{s, s_4\}$.

Dans le cas d'un système de voisinage à 8 (chaque site a exactement huit voisins), il y a 8 cliques d'ordre 2, 12 cliques d'ordre 3 et 4 cliques d'ordre 4 [35].

S ₁	S ₂	S ₃
S ₄	s	S ₅
S ₆	S ₇	S ₈

Cliques du 8-voisinage:

- Les singletons.
- Cliques Ordre 2: $\{s, s_1\}$; $\{s, s_2\}$; $\{s, s_3\}$; $\{s, s_4\}$; $\{s, s_5\}$; $\{s, s_6\}$; $\{s, s_7\}$; $\{s, s_8\}$
- Cliques Ordre 3: $\{s, s_1, s_2\}$; $\{s, s_2, s_3\}$; $\{s, s_3, s_5\}$; $\{s, s_5, s_8\}$; $\{s, s_8, s_7\}$; $\{s, s_7, s_6\}$; $\{s, s_6, s_4\}$; $\{s, s_4, s_1\}$; $\{s, s_2, s_4\}$; $\{s, s_2, s_5\}$; $\{s, s_5, s_7\}$; $\{s, s_7, s_4\}$.
- Cliques Ordre 4: $\{s, s_1, s_2, s_4\}$; $\{s, s_2, s_3, s_5\}$; $\{s, s_5, s_8, s_7\}$; $\{s, s_7, s_6, s_4\}$.

2-3-1-3- Définition d'un champ de Markov [14]:

- Les probabilités conditionnelles de la forme :

$$P(X_F = x_F / X_{\Omega-F} = x_{\Omega-F}) \quad (3.11)$$

sont appelées caractéristiques locales.

Où $F \subset \Omega$ et X_F est la restriction du champ X à F .

• Un vecteur aléatoire $X = (X_s, s \in \Omega)$ est un champ de Markov relativement à un système de voisinage V si et seulement si:

- 1) $\forall x \in \mathcal{X}, P(x) > 0$
- 2) $P(X_s = x_s | (X_r = x_r), r \in \Omega - \{s\}) = P(X_s = x_s | (X_r = x_r), r \in V_s)$

Ce qui signifie que la connaissance d'un voisinage local du pixel s est suffisante pour définir la loi conditionnelle de X_s .

2-3-1-4- Potentiel de Gibbs et champ de Gibbs [54][22]:

• Un potentiel de Gibbs est une famille $U = \{U_F, F \subset \Omega\}$ de fonctions sur \mathcal{X} telles que:

- 1) $U_\emptyset = 0$
- 2) $U_F(x) = U_F(y)$ si $x_F = y_F$

Où x_F est la restriction de la configuration x sur les sites de F .

• L'énergie associée au potentiel U est donnée par:

$$H_U(x) = \sum_{F \subset \Omega} U_F(x) \quad (3.12)$$

• Le potentiel U définit une loi de probabilité sur \mathcal{X} :

$$P(x) = \frac{\text{Exp}[-H_U(x)]}{Z(U)} \quad \text{avec} \quad Z(U) = \sum_{y \in \mathcal{X}} \text{Exp}[-H_U(y)] \quad (3.13)$$

• La loi de probabilité définie en (3.13) est appelée **champ de Gibbs** ou **mesure de gibbs** engendrée par la fonction énergie H .

2-3-2- Aspects théoriques de l'algorithme Recuit 3:

On considère Ω un ensemble de N sites représentant la matrice image (ensemble de pixels). La collection $d = (d_1, \dots, d_N)$ où $d_i \in E = [0, 255] \cap \mathbb{N}$, représente l'image initiale observée.

De même que l'algorithme Recuit2, cet algorithme procède à l'affectation des pixels (s_1, \dots, s_N) de l'image dans M classes C_m , $m = 1, \dots, M$, caractérisées chacune par un niveau de gris a_m .

Nous modélisons le phénomène aléatoire qui affecte un pixel s de Ω , dans une classe de niveau de gris a_m , par une variable aléatoire X_s à valeurs dans $\Phi = \{a_1, \dots, a_M\}$. Le phénomène d'apparition d'une configuration de l'espace $\chi = \Phi^N$ est alors représenté par un champ aléatoire $X = (X_s)_{s \in \Omega}$, constitué de la collection des variables X_s .

La loi de probabilité de X , notée $P = (P(X=x))_{x \in \chi}$ représente la loi jointe du champ X . C'est la fréquence d'apparition des différentes configurations de l'espace χ .

L'objectif de Recuit3 est de retrouver une configuration $x = (x_1, \dots, x_N)$ (réalisation du champ aléatoire X), qui soit une solution optimale d'une fonction objective que l'on définira ultérieurement. En termes statistiques, il s'agit de faire une estimation de la configuration x .

Plusieurs méthodes d'estimation sont envisageables. La méthode la plus communément utilisée est la maximisation de la probabilité a posteriori $P(x|d)$, où d est l'image observée. Cette maximisation n'est pas réalisable directement, vu que l'espace des configurations χ est trop grand. L'exploration de toutes les configurations et le calcul de toutes les probabilités $P(x|d)$ requiert un temps d'exécution et un espace mémoire très grand. En effet, il s'agit de déterminer N valeurs (x_1, \dots, x_N) , chaque valeur peut être choisie parmi M valeurs de $\Phi = \{a_1, \dots, a_M\}$ et où chaque a_m est pris dans E .

Afin de s'affranchir de cette lourde tâche, nous traitons l'image d'une manière locale. Le champ aléatoire X ne sera plus caractérisé par sa loi jointe $P(X=x)$, mais par ses lois marginales sur des petites parties de l'ensemble Ω . On suppose alors, que les caractéristiques locales extraites sur ces différentes parties, par la réalisation du champ X sont équivalentes à celles extraites sur plusieurs configurations de toute l'image [14].

2-3-2-1- Modélisation locale du champ X :

Le champ aléatoire X sera caractérisé partiellement, par ses lois marginales sur des sous ensembles de Ω . On définit pour chaque pixel s de Ω , son voisinage à 4 V_s et on suppose que X est Markovien:

$$\forall s \in \Omega, P(X_s = x_s | (X_r = x_r), r \in \Omega - \{s\}) = P(X_s = x_s | (X_r = x_r), r \in V_s) \quad (3.14)$$

Cela permet d'expliciter la loi globale du champ X à partir de ses lois conditionnelles (3.14) définies sur chaque pixel s de Ω sachant l'observation sur le voisinage de s .

2-3-2-2- Hypothèse de stationnarité:

On suppose que les lois conditionnelles $P(X_s | (X_r), r \in V_s)$ sont indépendantes de la localisation géométrique de s sur le plan de l'image Ω .

2-3-2-3- Représentation du champ de Markov X par une distribution de Gibbs:

Le théorème de Hammersley-Clifford [8][56], montre qu'un champ de Markov X défini sur un espace Ω relativement à un système de voisinage V, peut être défini par un champ de Gibbs de distribution de probabilité:

$$P(x) = \frac{1}{Z} \text{Exp} -H(x) \quad (3.15)$$

Où

Z est appelée fonction de partition du système. C'est une constante définie par :

$$Z = \sum_{x \in \mathcal{X}} \text{Exp}(-H(x)) \quad (3.16)$$

On note que Z est incalculable si $|\mathcal{X}|$ est grand.

H(x): Fonction Energie globale du système, définie par:

$$H(x) = \sum_{C \in \zeta} U_C(x_C) \quad (3.17)$$

H s'exprime comme une somme sur le nombre de cliques de Ω , de fonctions $U_C(x_C)$.

ζ : ensemble de cliques correspondant au système de voisinage V.

x_C : restriction du champ X à la clique C.

$U_C(x_C)$: Fonction potentiel, permettant d'exprimer les contraintes du voisinage. Elle dépend uniquement des éléments x_s tels que s appartient à la clique C.

Donc si on connaît les valeurs de la fonction potentiel U_C sur l'ensemble des réalisations de la restriction X_C à chaque clique C, on pourra simuler le champ global X. Ceci est réalisé en explicitant les lois conditionnelles de X (définies sur chaque pixel s):

$$P(X_s = v | (X_r = x_r), r \in \Omega - \{s\}) = \frac{P(X_s = v, (X_r = x_r), r \in \Omega - \{s\})}{\sum_{w \in \Phi} P(X_s = w, (X_r = x_r), r \in \Omega - \{s\})} \quad (3.18)$$

$$= \frac{\text{Exp} [- \sum_{C/s \in C} U_C(x_{|_{x=v}})]}{\sum_{w \in \Phi} \text{Exp} [- \sum_{C/s \in C} U_C(x_{|_{x=w}})]} = P(X_s = v | (X_r = x_r), r \in V_s) \quad (3.19)$$

Où $x_{|_{x=w}}$ signifie que $x_s = w$ et les valeurs aux sites r distincts de s sont les x_r .

On note que dans l'équation (3.19), seule la somme sur les $w \in \Phi$ est calculée et non la somme sur l'espace \mathcal{X} des configurations, ce qui simplifie les calculs.

L'équation (3.19) associe aux probabilités conditionnelles $P(X_s = \nu \mid (X_r = x_r), r \in V_s)$, des énergies conditionnelles :

$$H(X_s \mid (X_r), r \in V_s) = \sum_{C/s \in C} U_C(x_C) \tag{3.20}$$

D'où, le choix d'un état x_s pour le pixel s , peut être aussi bien mesuré par H que par la probabilité P . Une valeur minimale de H correspond à une valeur maximale de P .

2-3-2-4- Simulation du champ X:

Nous choisissons une forme de Gibbs du champ X , correspondant au choix du 4-voisinage et des cliques associées. Pour un pixel s , son voisinage est $V_s = \{r_1, r_2, r_3, r_4\}$. Les cliques correspondantes sont les singletons et les cliques d'ordre 2: $C_1 = \{s, r_1\}$, $C_2 = \{s, r_2\}$, $C_3 = \{s, r_3\}$, $C_4 = \{s, r_4\}$.

	r_1	
r_2	s	r_4
	r_3	

A chaque clique C_k , on associe une fonction potentiel U_{C_k} dépendant uniquement des éléments de C_k (ie. s et r_k).

$$U_{C_k} = \begin{cases} 0 & \text{si } |C_k|=0 \\ (x_s - x_{r_k})^2 + 1/4 (x_s - d_s)^2 & \text{si } k=1 \dots 4 \end{cases} \tag{3.21}$$

D'où

$$H(X_s = a \mid (X_r), r \in V_s) = (a - d_s)^2 + \sum_{k=1}^4 [(a - x_{r_k})^2] \tag{3.22}$$

• **Justification du choix de H:**

La fonction énergie associée à un pixel s sachant l'observation de son voisinage, s'exprime comme une somme de deux termes:

Le premier terme représente l'écart quadratique entre le niveau de gris a proposé pour s et son niveau de gris initial d_s . La minimisation de ce terme permet de retenir le niveau de gris a qui soit le plus proche possible de l'état observé d_s sur l'image initiale.

Le second terme est sous forme d'une somme des écarts quadratiques entre le niveau de gris proposé a et les niveaux de gris des pixels voisins de s . La minimisation de ce terme exprime l'homogénéité d'un voisinage.

2-3-2-5- Recherche de la configuration optimale du champ X:

La minimisation de l'énergie H exprimée en (3.22) permet d'attribuer à un pixel s , un nouvel état $x_s^{(t+1)} = a_{\min}$. La synthétisation d'une nouvelle configuration x du champ X , se fait en balayant toute l'image ligne par ligne et en déterminant pour chaque pixel son nouvel état minimisant l'énergie locale H de (3.22).

L'algorithme du recuit simulé [31] permet de retrouver la configuration optimale x minimisant la fonction énergie globale $H(x)$ de l'équation (3.17).

• Schéma général du recuit simulé avec l'Echantillonneur de Gibbs [14]:

- Tirer une configuration aléatoire $x^{(0)}$
- Fixer une température initiale θ_0 et poser $\theta = \theta_0$
- Balayer l'ensemble des sites S (dans n'importe quel ordre). Chaque site s visité à l'étape t est caractérisé par son état $x_s^{(t)}$ et par les configurations $x_{V_s}^{(t)}$ de son voisinage $V_s(t)$. Son nouvel état $x_s^{(t+1)}$ est obtenue de la manière suivante:
 - Calculer les probabilités conditionnelles : $P(\cdot | x_r^{(t)}, r \in V_s^{(t)})$
 - Effectuer un tirage aléatoire de $x_s^{(t+1)}$ suivant cette loi.
- Faire décroître la température θ_t
- Poursuivre les balayages jusqu'à la réalisation d'un critère d'arrêt.

Algorithme 3.10 : Recuit simulé avec échantillonneur de Gibbs.

2-3-2-6- Algorithme Recuit 3:

Début

Lecture de la température initiale θ_0 et de la température seuil θ_{\min} .

Lecture de l'image initiale: $d(s_1, \dots, s_N) = (d_1, \dots, d_N)$;

$t \leftarrow 0$, Choisir aléatoirement : $\Phi = \{a_1, a_2, \dots, a_M\}$;

Choisir aléatoirement $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)})$ dans Φ ;

1- Pour $i \leftarrow 1$ à N

faire (*choisir pour le pixel s_i un nouvel état $x_i^{(t+1)} = a_m$ tel que a_m engendre une énergie minimale pour s_i *)

$$H_{a_{\min}}^{(t)}(s_i) = \underset{m=1..M}{\text{Min}}(H_{a_m}(s_i)) = \underset{m=1..M}{\text{Min}} [(a_m - d_i)^2 + \sum_{k=1}^4 (a_m - x_{r_k}^{(t)})^2]$$

$x_i^{(t+1)} \leftarrow a_{\min}$

Fait

$\theta_{t+1} \leftarrow 0.95 * \theta_t$

Si $\theta_t < \theta_{\min}$ alors Fin

sinon $t \leftarrow t+1$; aller à 1

fsi;

Fin;

Algorithme 3.11 : Recuit 3.

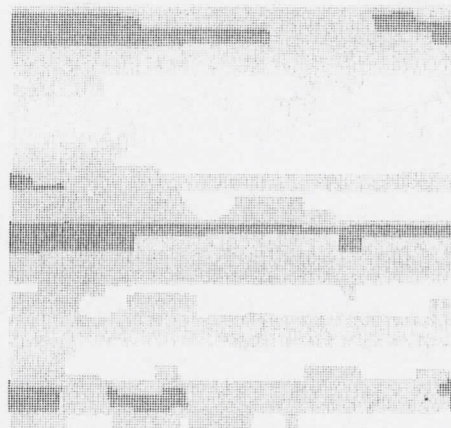
2-3-3- Application de Recuit 3:

Données: L'algorithme Recuit3 est appliqué sur des fenêtres de l'image SPOT XS1, de résolution spatiale 20x20m et représentant la région de l'Atlas Blidéen. En faisant varier les paramètres température initiale et température minimale.

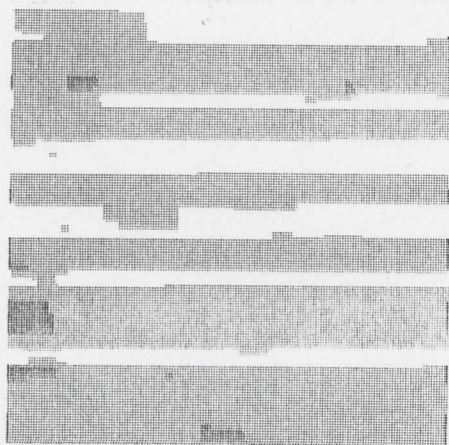
2-3-3-1- Affichage des résultats:



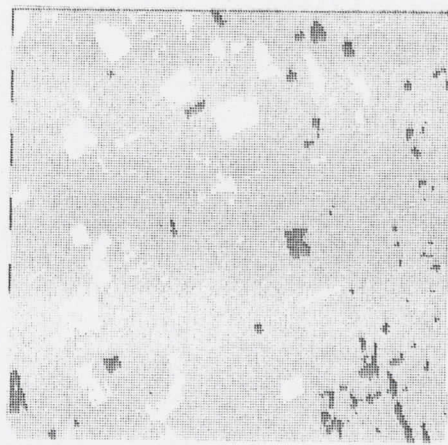
1-) Image originale
170 x 170



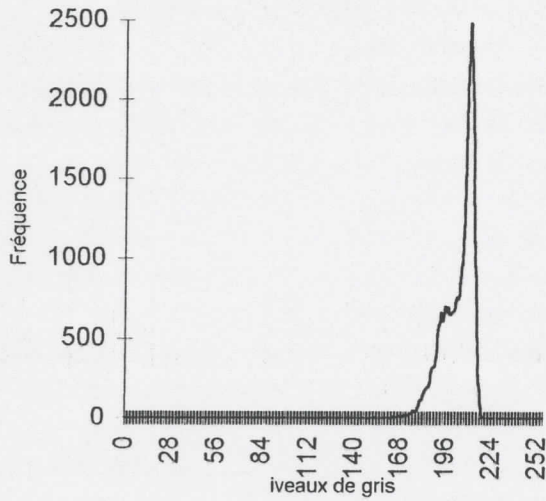
2-) Image résultat
Cas a: $\theta_o = 100$ $\theta_{min} = 1$ $M = 3$



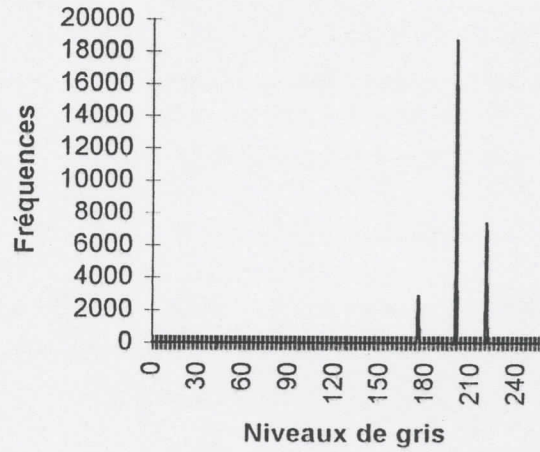
3-) Image résultat
Cas b: $\theta_o = 10$ $\theta_{min} = 0.01$ $M = 3$



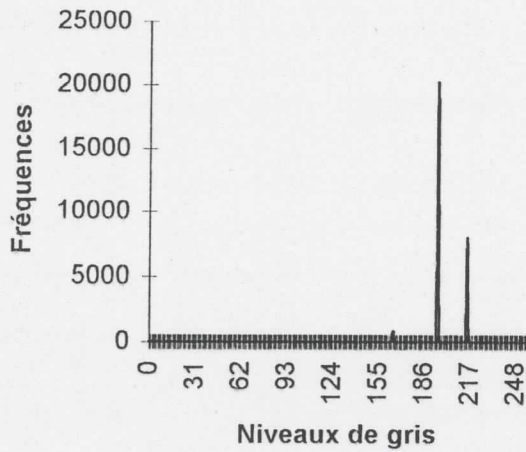
4-) Image résultat
Cas c: $\theta_o = 1000$ $\theta_{min} = 0.01$ $M = 3$



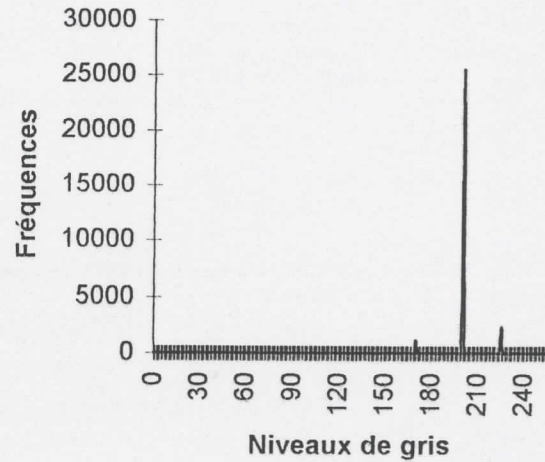
5-) Histogramme de (1)



6-) Histogramme de (2)



7-) Histogramme de (3)



8-) Histogramme de (4)

Figure 3.8 : Application de Recuit3. Images et Histogrammes.

2-3-3-2- Interprétation des résultats de l'algorithme Recuit 3:

Cas	Nbre de classes M	θ_0	θ_{min}	a_1	a_2	a_3	Nombre d'itérations	Température finale	Energie finale
a-)	3	100	1	176	201	221	215	0.99	0.29
b-)	3	10	0.01	165	195	215	650	0.009	0.16
c-)	3	1000	0.001	171	200	226	7500	0.0009	0.113

Tableau 3.3: Résultats de l'application de Recuit3.

Le rôle de l'algorithme Recuit3 est d'affecter les pixels de l'image dans des classes de niveau de gris déterminé. Partant d'une configuration initiale de l'image, choisie aléatoirement dans l'espace des configurations, l'algorithme génère par le simulateur de Gibbs de nouvelles configurations en attribuant aux pixels de l'image des valeurs de gris caractérisant les différentes classes. Ce choix doit maximiser les probabilités conditionnelles (3.19) caractérisant le champ de Markov X . Cela correspond à la minimisation de la fonction énergie décrite en (3.22). La génération des configurations se fait, d'une manière progressive, par recuit simulé.

On fait varier, dans cette application, les paramètres température. La figure 3.8 cas (a) et (b), montre que pour les intervalles de température $[100, 1]$ et $[10, 0.01]$, correspondant à un nombre de balayages de l'image (itérations) inférieur à 7500 (Tableau 3.3), les images obtenues présentent des plages de 3 niveaux de gris, ne correspondant pas à l'image initiale.

Dans le cas (c), on fixe l'intervalle de température à $[1000, 0.001]$. Sur l'image obtenue (Figure 3.8 image 4), on remarque l'apparition de zones homogènes de 3 niveaux de gris différents correspondant aux 3 classes C_m $m=1..3$.

2-4- Comparaison des résultats des 3 modèles :

Le tableau suivant résume les résultats obtenus dans les cas d'application retenus, pour chacun des trois algorithmes proposés:

<i>Méthodes</i>	<i>Intervalle de température</i>	<i>Analyse visuelle de l'image résultat</i>	<i>Espace mémoire (octets)</i>	<i>Nombre d'itérations</i>
Recuit1(casb)	10 - 0.01	- Mise en évidence de quelques zones homogènes.	30000	650
Recuit2(casb)	10^4 - 0.01	- Image sous forme de bandes de 3 niveaux de gris différents correspondant aux trois classes.	30150	3400
Recuit3(casc)	10^3 - 10^{-3}	- Apparition d'un nombre plus importants de zones homogènes.	30100	7500

Tableau 3.4: Tableau comparatif des résultats du Recuit simulé.

L'algorithme Recuit1, qui consiste à rechercher la configuration optimale de l'histogramme de l'image, a donné des résultats moyens pour un nombre d'itérations faible. Le Recuit2 qui tend à retrouver la configuration optimale de l'image elle même, a donné des résultats aberrants. Le Recuit3, qui consiste lui aussi à retrouver la configuration optimale de l'image mais en y effectuant un traitement local, a donné de meilleurs résultats pour un nombre d'itérations plus élevé.

CHAPITRE 4:

L'APPROCHE CONNEXIONISTE EN SEGMENTATION D'IMAGES

- 1- Généralités sur les Réseaux de Neurones Artificiels.
- 2- Modèle de Kohonen.
- 3- Modèle de Kohonen en Segmentation d'images.

Chapitre 4: **APPROCHE CONNEXIONISTE EN SEGMENTATION D'IMAGES**

1- Généralités sur Les réseaux de neurones artificiels:

L'approche connexioniste, ou approche des réseaux de neurones s'inspire des modèles de comportement des mécanismes de base du cerveau humain. Son idéal est de retrouver le réflexe des êtres vivants qu'est la réaction vis à vis d'une certaine situation. Cette situation peut être perçue par l'ouïe, la vue, l'odorat, le toucher et le goût. De ces cinq sens, nous nous intéressons à la perception visuelle. Celle-ci peut être définie simplement par le fait de se rendre compte de la présence d'un signal. Cependant à un stade plus avancé, la perception va jusqu'à répondre d'une manière appropriée à ce signal. Pour cela, le signal doit subir plusieurs phases de traitement. Dans ce chapitre, nous nous limiterons à l'étape de la segmentation des images, celle-ci sera modélisée par l'approche connexioniste des réseaux de neurones artificiels (R.N.A.).

Les R.N.A. [4][11] sont des modèles mathématiques imitant la structure et les fonctions des réseaux de neurones biologiques. Un R.N.A est composé de plusieurs cellules appelées *neurones* ou *unités de calcul*, liées entre elles par des connexions affectées de *poids* w_{ij} , où j est le neurone de départ et i est le neurone d'arrivée. Ces connexions permettent la transmission des informations entre les neurones. Les cellules d'un R.N.A. sont de trois types, les cellules d'entrée, les cellules intermédiaires et les cellules de sortie.

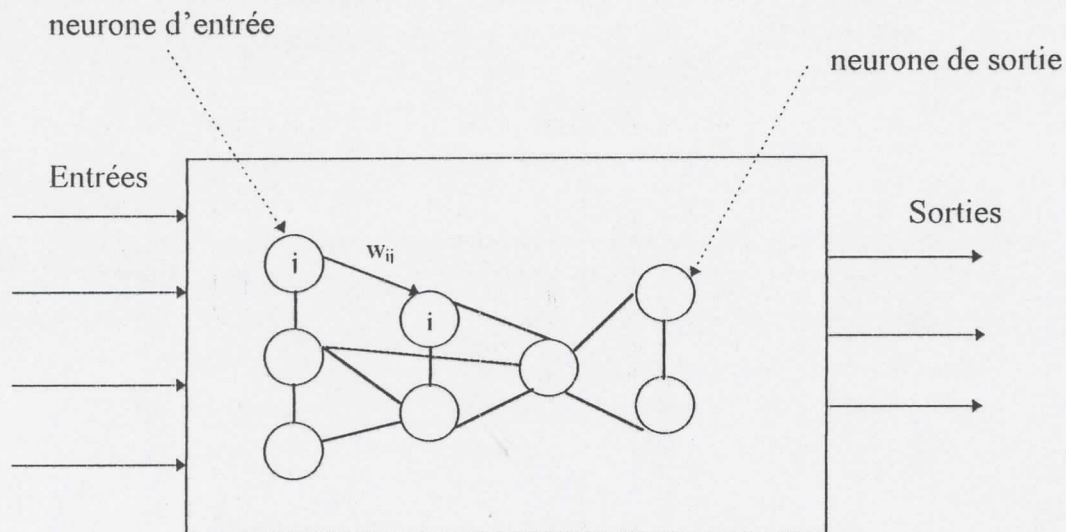


Figure 4.1 : Schéma d'un RNA.

Un RNA peut être considéré comme étant une fonction $F_w: R^n \longrightarrow \xi^p$, paramétrée par les poids w_{ij} des connexions où

- n est le nombre de cellules d'entrée,
- p est le nombre des unités de sorties,
- R est l'ensemble des valeurs en entrée,
- ξ est l'ensemble des valeurs des cellules de sortie.

Pour des formes de R^n (entrées), le réseau renvoie en réponse des formes correspondantes de ξ^p .

Historiquement, l'approche des RNA trouve ses origines dans les convictions des chercheurs qui y sont intéressés.

En 1943, McCulloch et Pitts [36] étudièrent un ensemble de neurones formels interconnectés pouvant calculer certaines fonctions logiques.

En 1949, Hebb [25] montre l'importance du couplage synaptique dans les processus d'apprentissage.

En 1958, Rosenblatt [45], partant des pensées de Hebb, McCulloch et Pitts, met au point le premier modèle opérationnel de réseau de neurones: **le perceptron**.

Ce modèle a suscité beaucoup de recherches dans cet axe, jusqu'à 1969, où les deux mathématiciens Minsky et Paper [38] ont montré les limites théoriques du perceptron. Ceci a induit un recul dans les réseaux de neurones.

En 1982, les réseaux de neurones ont connu une renaissance remarquable, grâce à Hopfield, qui a permis d'apporter une assise théorique aux réseaux neuronaux.

En 1985, de nouveaux modèles mathématiques ont permis de dépasser les limites du perceptron [32].

1-1- Caractéristiques des Réseaux de Neurones Artificiels (RNA):

Un RNA est défini par :

- Ses unités de calcul.
- Son architecture.
- Le protocole de communication entre neurones (règle d'apprentissage et règle d'activation).

1-1-1- Les unités de calcul (neurones):

Un neurone est un automate capable d'effectuer des opérations mathématiques simples. Il possède plusieurs connexions entrantes lui délivrant différentes informations issues des neurones voisins et plusieurs connexions en sortie présentant toutes la même réponse [4],[50].

Chaque neurone N_i reçoit n signaux en entrée constituant un vecteur $e_i = (e_i^{(1)}, e_i^{(2)}, \dots, e_i^{(m)})$ appelé quantité d'activation en entrée du neurone N_i . Il combine les éléments $e_i^{(1)}, e_i^{(2)}, \dots, e_i^{(m)}$ avec les poids respectifs w_{ij} $j=1, \dots, m$ de chaque connexion, pour former la fonction d'entrée. Il applique une fonction d'activation $a_i = f(e_i^{(j)}, w_{ij})_{j=1, \dots, m}$ à cette fonction d'entrée afin de mettre à jour son niveau d'activation.

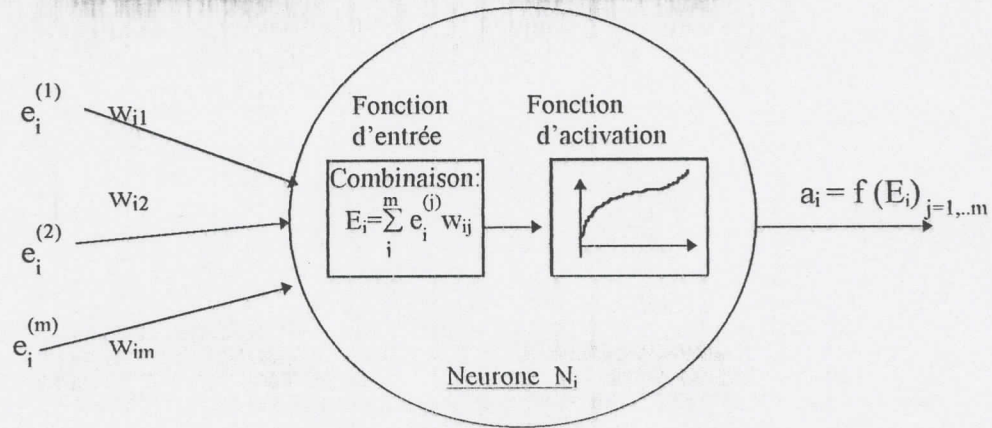


Figure 4.2: Schéma d'un neurone artificiel.

- La fonction d'entrée combine le vecteur d'entrée e_i et les poids des connexions entrantes. Elle peut être une sommation comme dans l'exemple (figure 4.2) ou une fonction non linéaire.

Exemple:
$$E_i = \sum_j^m e_i^{(j)} w_{ij} - \theta \quad \text{avec } \theta \in \mathbb{R}$$

- La fonction d'activation est une procédure locale que chaque noeud exécute en mettant à jour son niveau d'activation en fonction du contexte de son voisinage. On utilise généralement trois types de fonctions d'activation:

- Les fonctions à seuils:

Exemples:

1) fonction à seuil linéaire:

$$f(E_i) = \begin{cases} 1 & \text{si } E_i > \theta \\ 0 & \text{si } E_i < \theta \end{cases} \quad \text{avec } \theta \in \mathbb{R}$$

2) fonction à seuil non linéaire:

$$f(E_i) = \begin{cases} 0 & \text{si } E_i < \theta_1 \\ \alpha E_i + \beta & \text{si } E_i \in [\theta_1, \theta_2] \\ 1 & \text{si } E_i > \theta_2 \end{cases} \quad \text{avec } (\theta_1, \theta_2, \alpha, \beta) \in \mathbb{R}^4$$

- Les fonctions linéaires:

1) $f(E_i) = \alpha E_i$ avec $\alpha \in \mathbb{R}$

- **Les fonctions sigmoïdes:** ce sont des fonctions à seuil, molles comme la tangente hyperbolique.

Exemples:

$$1) f(E_i) = 1/(1 + \exp(-E_i))$$

$$2) f(E_i) = \left[1 + \exp\left(-\frac{(E_i - \theta_1)}{\theta_2}\right) \right]^{-1} \quad \text{avec } (\theta_1, \theta_2) \in \mathbb{R}^2$$

1-1-2- Architecture des réseaux: C'est la manière de concevoir les réseaux, elle est spécifiée par le nombre de neurones, la nature des neurones (fonctions d'entrée et d'activation), le graphe d'interconnexions des cellules et la relation entre le réseau et l'extérieur.

L'architecture des RNA varie d'une connectivité totale à une connectivité locale où les neurones ne sont connectés qu'à leurs voisins mitoyens.

Dans les réseaux de neurones artificiels, il existe trois types de cellules:

- Cellules d'entrées formant la rétine du réseau.
- Cellules cachées ou cellules d'association.
- Cellules de sortie ou cellules de décision.

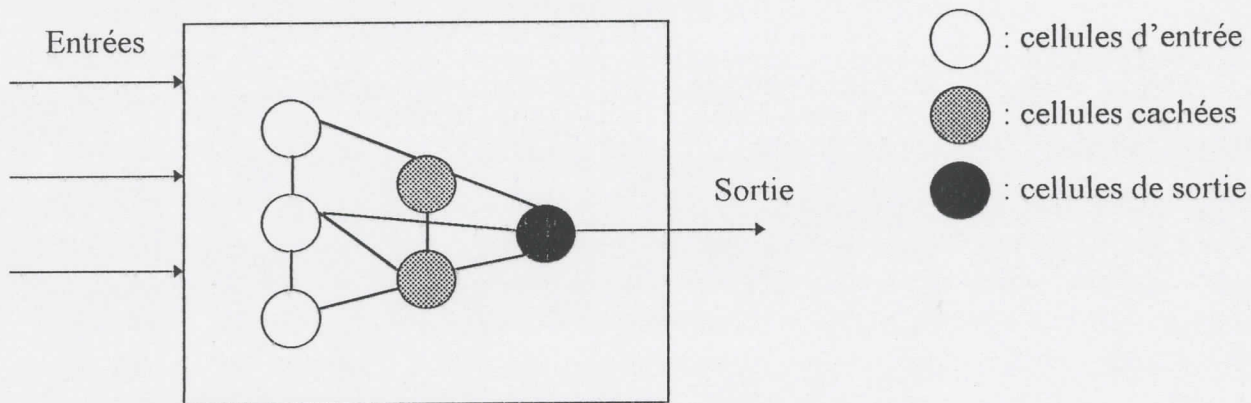


Figure 4.3 : Structure d'un RNA

-Types d'architecture de réseaux:

1- Les réseaux monocouches :

Les réseaux monocouches présentent une couche de cellules d'entrée, une couche de cellules de sortie et une seule couche de cellules cachées, d'où leur appellation. Le premier modèle de réseaux monocouches appelé Perceptron a été développé par ROSENBLATT en 1957 [4].

2- Les réseaux multicouches : Ils comportent une couche en entrée, une couche délivrant les sorties et plusieurs couches intermédiaires appelées couches cachées. Les neurones d'une même couche ne sont pas connectés entre eux.

2- Les réseaux à connexion entière: Chaque neurone est connecté à lui-même et à tous les neurones du réseau.

1-1-3- Protocole de communication:

1-1-3-1- Mécanisme d'apprentissage:

Les psychologues définissent l'apprentissage comme étant un changement dans le comportement d'un sujet par rapport à une situation donnée, changement provoqué par ses expériences répétées, dans cette situation.

Dans le cerveau des êtres vivants, l'apprentissage est caractérisé au niveau biochimique par l'évolution des connexions synaptiques entre les neurones.

Dans les RNA, les mécanismes d'apprentissage sont des procédures locales décrivant la modification des poids des connexions entre les neurones en fonction du temps [11].

L'apprentissage dans les réseaux de neurones est de trois types:

1- Apprentissage supervisé: On présente au réseau, un ensemble d'apprentissage E de N paires (x_k, y_k) avec x_k : entrée et y_k : sortie associée. Le RNA calcule pour chaque entrée x_k la sortie s_k correspondante, si la sortie calculée ne correspond pas à la sortie théorique y_k , le réseau met à jour les poids des connexions et reprend les calculs.

Le RNA doit être capable de mettre toutes les formes (x_k, y_k) en correspondance et ceci par :

- choisir l'architecture,
- Déterminer un ensemble de poids w permettant à cette architecture de réaliser l'association voulue (x_k, y_k) .

2- Apprentissage guidé: Dans ce cas, on indique au RNA si la sortie s_k produite pour une entrée x_k est correcte ou pas, par un signal d'échec

3- Apprentissage non supervisé: Dans un apprentissage non supervisé, on ne demande pas explicitement au réseau d'associer un y_k à chaque x_k . Les formes x_k sont présentées au réseau et celui-ci doit s'auto-organiser afin de produire des sorties proches en réponse à des entrées proches. L'appréciation de la qualité de l'apprentissage est généralement laissée au gré du concepteur.

1-1-3-2- Règle de HEBB: [3]

Si deux neurones connectés entre eux sont activés en même temps, alors la connexion qui les relie doit être renforcée, dans le cas contraire, elle n'est pas modifiée (où certaines fois diminuée).

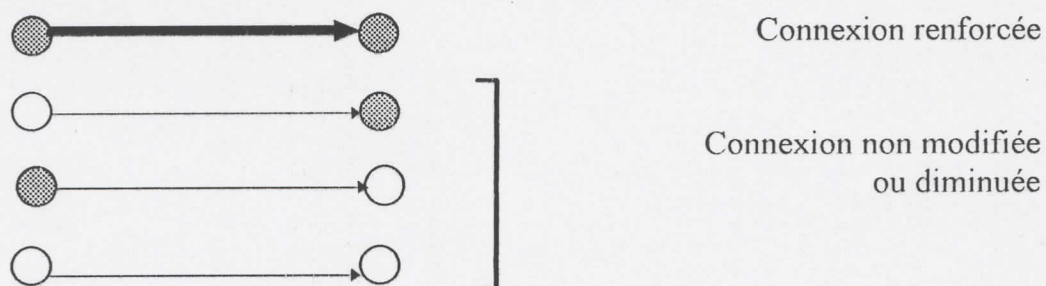


Figure 4.4: Règle de HEBB.

2- Modèle de KOHONEN:

En 1985, Kohonen établit un nouveau modèle de réseau multicouches dont les neurones sont activés par une fonction sigmoïde [32].

Le modèle de KOHONEN procède d'une observation neurophysiologique du cortex cérébral. KOHONEN fit trois constatations [3][11].

1- Phénomène de réponse localisée: Un certain stimuli engendre toujours la même réaction de la part des mêmes cellules nerveuses. C'est le phénomène de réponse localisée, par conséquent, certaines cellules nerveuses ont un rôle bien précis dans le système de perception.

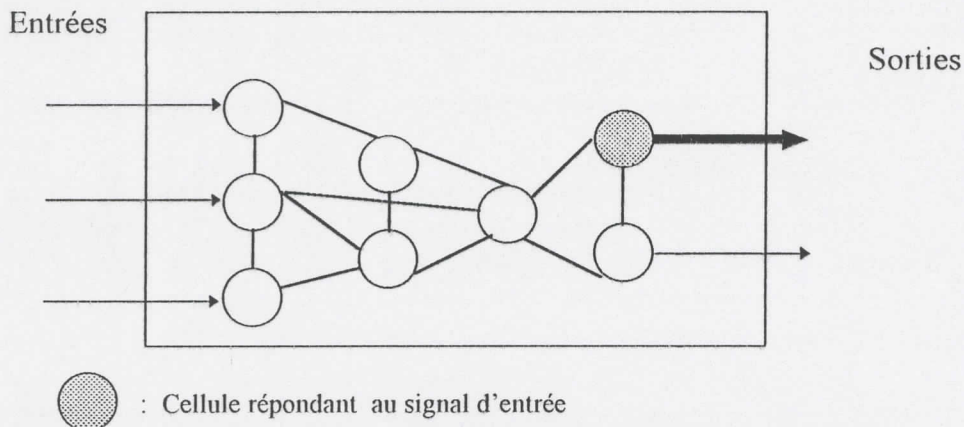


Figure 4.5: phénomène de réponse localisée.

2- Interaction interneuronale: La deuxième constatation est qu'il existe un mécanisme d'interaction latérale entre les neurones recevant les signaux. Cette interaction dépend de la distance entre ces neurones. Cette dépendance a été schématisée par la fonction du chapeau mexicain.

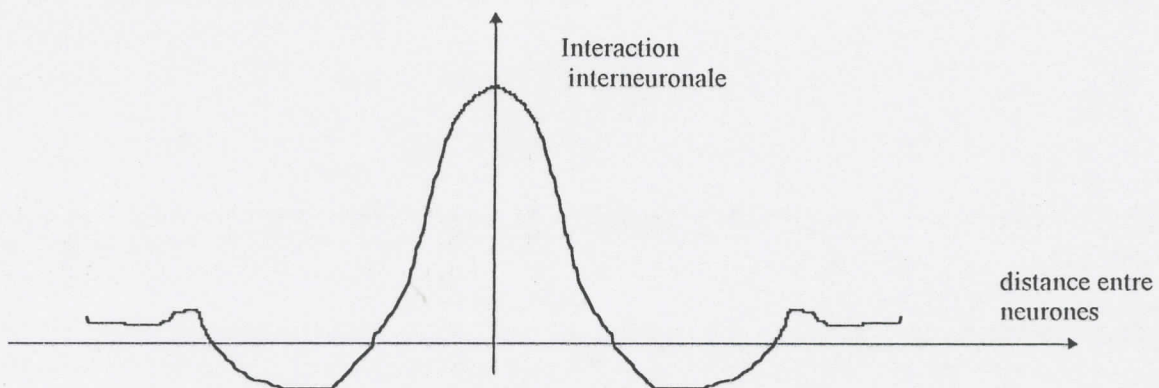


Figure 4.6: Interaction latérale entre neurones.

3- Activation des neurones: Si un neurone reçoit un potentiel d'action par une de ses connexions et qu'il s'active, il renforce cette connexion. S'il ne reçoit aucun potentiel ou si le potentiel arrivant par une certaine connexion n'est pas assez fort pour déclencher une activité. Cette connexion reste inchangée.

Partant de ces observations sur les systèmes nerveux naturels, Kohonen établit son modèle de Réseaux de neurones artificiels.

2-1- Type de neurones du réseau de Kohonen:

Les neurones utilisés sont à fonction d'activation sigmoïde: $f(x) = 1/(1+\exp(-x))$

2-2- Architecture du réseau de Kohonen:

Les neurones de ce modèle sont regroupés en couches:

- Une couche d'entrée définissant l'espace des entrées possibles.
- Une grille de neurones de sortie dite couche compétitive. Chaque neurone de la couche d'entrée est connecté à tous les neurones de la couche compétitive [50].

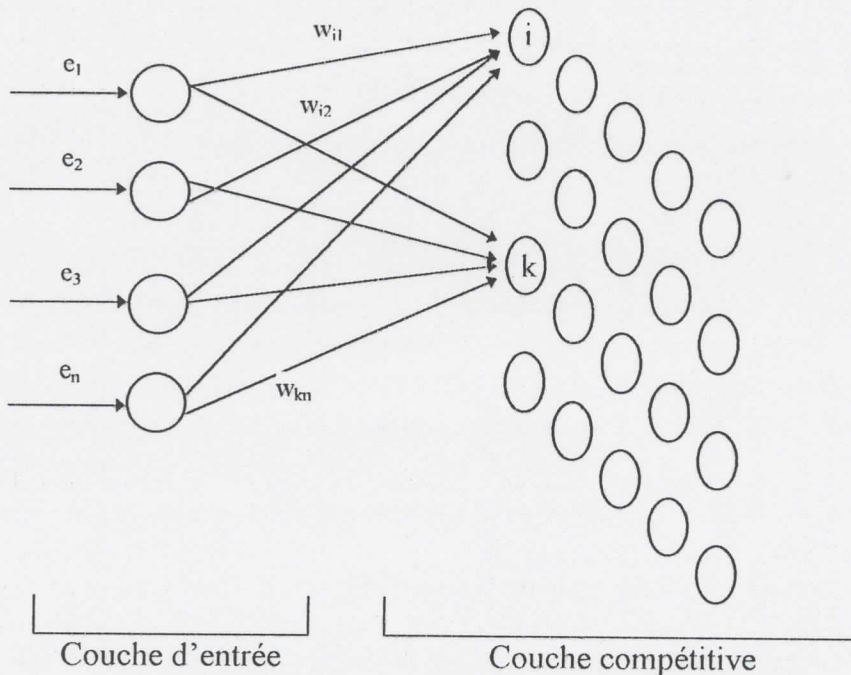


Figure 4.7- Structure du réseau de Kohonen.

Le vecteur d'entrée est défini par: $e = (e_1, e_2, \dots, e_n)$. Les poids des connexions sont notés w_{ij} où i est un neurone de la couche d'entrée et j un neurone de la couche compétitive.

2-3- Règle d'apprentissage du modèle de Kohonen:

En s'appuyant sur le phénomène de la réponse localisée observé sur les systèmes nerveux biologiques, Kohonen propose une règle d'apprentissage non supervisée; les entrées arrivent aux cellules de la couche compétitive via les cellules de la couche d'entrée. Les unités de la couche compétitive se disputent la possibilité de répondre à un signal d'entrée donné et un seul neurone doit s'activer. C'est le phénomène de tout au vainqueur, « Winner takes all » correspondant au phénomène de réponse localisée des systèmes biologiques. La concurrence entre les neurones de la couche compétitive peut être établie de deux façons[11]:

1) Sélectionner le neurone vainqueur présentant la plus faible distance entre le signal d'entrée et le poids de sa connexion:

$$d_{N_v} = \min_i d_{N_i} = \min_i \| e - w_i \| = \min_i \sqrt{\sum_j (e_j - w_{ij})^2} \quad (4.1)$$

d_{N_v} : distance du neurone vainqueur N_v .

2) Chaque cellule de la couche compétitive inhibe les connexions des autres neurones.

Généralement, le modèle de Kohonen utilise la première méthode de compétition qu'est la recherche du neurone gagnant.

Partant aussi, du phénomène d'interaction interneuronale, Kohonen propose de mettre à jours les poids des connexions des neurones appartenant au voisinage du neurone gagnant et de laisser les autres connexions inchangées.

Les poids des connexions seront ajustés par:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \alpha(t) [e_i(t) - w_{ij}(t)] & \text{si } j \in \text{voisinage de } N_v \\ w_{ij}(t) & \text{sinon} \end{cases} \quad (4.2)$$

$\alpha(t)$ est le taux d'apprentissage du réseau à l'itération t .

En se basant sur les équations (4.1) et (4.2), Kohonen propose un algorithme d'apprentissage itératif non supervisé:

1) Initialiser les poids par de faibles valeurs aléatoires.

2) Déterminer le neurone vainqueur par:

$$d_{N_v} = \min_i d_{N_i} = \min_i \| e - w_i \| = \min_i \sqrt{\sum_j (e_j - w_{ij})^2}$$

3) Ajustement des poids des connexions:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \alpha(t) [e_i(t) - w_{ij}(t)] & \text{si } j \in \text{voisinage de } N_v \\ w_{ij}(t) & \text{sinon} \end{cases}$$

4) $\alpha(t+1) = \epsilon \alpha(t)$ avec $\epsilon \in]0,1[$

Si $\alpha(t+1) >$ seuil alors aller à (2)

sinon arrêt

Algorithme 4.1: Algorithme d'apprentissage de Kohonen.

Le taux d'apprentissage du réseau $\alpha(t)$, permet d'apprécier les résultats de l'opération d'apprentissage.

3- Modèle de Kohonen en Segmentation des Images:

Le réseau de Kohonen est un modèle de classification en reconnaissance de formes, à apprentissage non supervisé, permettant de fournir des réponses correctes à des formes correctes [11].

Nous nous proposons d'utiliser ce modèle en segmentation des images. Rappelons que la segmentation des images consiste à affecter les pixels de l'image dans des classes caractérisées par un niveau de gris particulier. L'affectation d'un pixel à une classe est basée sur la valeur de son niveau de gris ainsi que les niveaux de gris de ses voisins.

L'objectif fixé est de concevoir un R.N.A. basé sur le modèle de Kohonen, dédié à la segmentation des images à niveau de gris et présentant une architecture et un algorithme d'apprentissage approprié.

Lors de l'étape de l'apprentissage, le réseau n'apprend pas des formes mais acquiert plutôt un pouvoir séparateur lui permettant d'estimer les différences des niveaux de gris des pixels appartenant à un même voisinage et décider de leur attribuer de nouvelles valeurs de gris.

3-1- Présentation du Réseau:

Le réseau de neurones dont nous exposons les caractéristiques dans ce qui suit, est basé sur le modèle de Kohonen utilisant une grille de compétition bidimensionnelle constituée de 256 unités de calcul. Chacune de ces unités représente un niveau de gris de l'échelle [0,255], et dispute la possibilité de répondre positivement à une entrée donnée.

3-1-1-Structure du Réseau :

Soient $G(L,K)$ la matrice des niveaux de gris de l'image à traiter et $O(L,K)$ la matrice de l'image résultat.

Couche d'entrée:

E est une couche bidimensionnelle de neurones d'entrée de taille $(L \times K)$.

Couche compétitive:

C est la couche compétitive du réseau. C'est une grille bidimensionnelle de taille $(16 \times 16 = 256)$. Chaque neurone représente une valeur de niveau de gris de la gamme [0,255].

Poids des connexions:

Pour chaque neurone i de la couche E , on définit une matrice W_i des poids des connexions de taille (16×16) , le reliant à toutes les unités de la couche C .

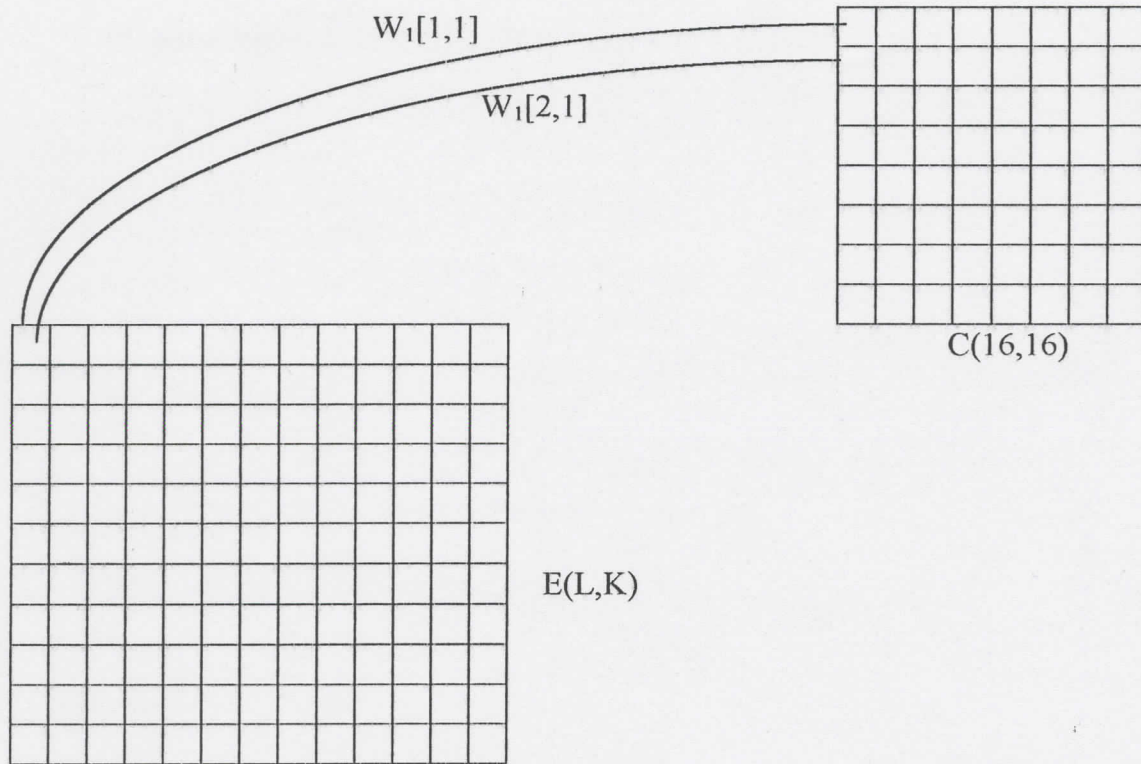


Figure 4.8 : Connexions du R.N.A.

3-1-2-Phase d'apprentissage:

Afin de préparer le réseau à des opérations de segmentation, il faut le soumettre à un protocole de communication permettant de déterminer les poids des différentes connexions.

Ce protocole débute par une initialisation aléatoire des poids des connexions reliant la couche E à la couche C.

On extrait de l'image A des fenêtres de taille (3x3) qui se recouvrent. Pour chacune de ces fenêtres de pixel central (i,j), on calcule la somme des différences entre son niveau de gris et les valeurs des niveaux de gris de ses voisins par:

$$E[i,j] = \sum_{u=i-1}^{u=i+1} \sum_{v=j-1}^{v=j+1} |G[i,j] - G[u,v]| \quad (4.3)$$

Pour tout neurone de la couche E, on calcule la distance entre sa valeur et les poids W le reliant aux unités de la grille C. L'unité de C représentant la distance minimale est retenue et la valeur de son poids ainsi que les valeurs des poids des deux unités voisines représentant les niveaux de gris respectivement supérieur et inférieur d'une unité sont mises à jour.

Le taux d'apprentissage est diminué. S'il ne satisfait pas la condition d'arrêt, on réitère l'opération d'apprentissage.

• **Algorithme d'apprentissage:**

Début

1) (* Introduction des paramètres du réseau *)

α : taux d'apprentissage.

seuil : seuil minimal du taux d'apprentissage.

fact_dim: Constante de diminution du taux α .

2) (* Initialisation du réseau *)

$t \leftarrow 0$;

Initialiser aléatoirement les poids de la matrice W par une valeur de [0,1]

Initialiser la couche E selon l'équation (4.3)

3) (* Etape itérative *)

Choisir une fenêtre (3x3) de E;

$t \leftarrow t+1$;

Déterminer k_1, k_2 tels que:

$d_{\min} \leftarrow \underset{k_1, k_2}{\text{Min}} (\text{abs}(E[2,2] - W[k_1, k_2]))$

$W[k_1, k_2] := W[k_1, k_2] + \alpha * (E[2,2] - W[k_1, k_2]);$

Si $k_2 > 1$ alors $W[k_1, k_2-1] := W[k_1, k_2-1] + \alpha * (E[2,2] - W[k_1, k_2-1]);$

Si $k_2 < 16$ alors $W[k_1, k_2+1] := W[k_1, k_2+1] + \alpha * (E[2,2] - W[k_1, k_2+1]);$

$\alpha := \alpha * \text{cste}$;

Si $\alpha < \text{seuil}$ alors aller à fin

sinon aller à étape 3;

Fin: Sauvegarde des poids W dans un fichier (poids.dat);

Algorithme 4.2: Phase d'apprentissage.

3-1-3- Phase de reconnaissance:

Le modèle de Kohonen est non supervisé, l'apprentissage est validé selon l'appréciation du concepteur. Une fois que la phase d'apprentissage est accomplie, des exemples d'images sont présentés au réseau. Si celui-ci ne répond pas correctement à ces exemples, l'opération d'apprentissage est relancé.

• Algorithme de reconnaissance:

Début

```

1) Chargement des poids du fichier "poids.dat".
2) Lecture de la taille de l'image (L,K).
3) Lecture des valeurs de gris de l'image.
4) Codage de l'image selon l'équation: (4.3)
5) Pour i ← 1 à L
   Faire Pour j ← 1 à K
     Faire (*Déterminer le neurone C[k1,k2], répondant positivement au pixel A[i,j]*)
       max ← Max ( (E(i,j)* W[k1,k2] )
                 k1,k2
       O[i,j] ← (k1-1)*16+k2-1;
     Fait;
   Fait;
sauvegarde de l'image résultat.
Fin;
```

Algorithme 4.3 : Phase de reconnaissance.

3-2- Application et Résultats:

Cas a:

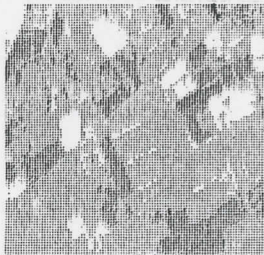

Paramètres du réseau		Image Initiale (1)	Image résultat (2)
α	= 100		
fact_dim	= 0.6		
seuil	= 1		
Nombre d'itération	= 100		
Nombre d'échantillons	= 9		

Figure 4.9 : Résultats de l'application Cas a.

Cas b:

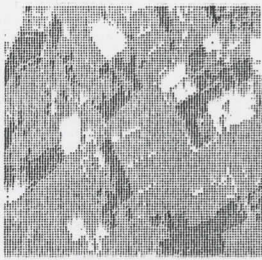
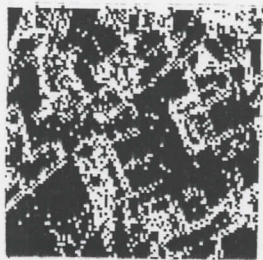
Paramètres du réseau		Image Initiale (1)	Image résultat (2)
α	= 10 000		
fact_dim	= 0.99		
seuil	= 1		
Nombre d'itération	= 26750		
Nombre d'échantillons	= 600		

Figure 4.10 : Résultats de l'application Cas b.

Cas c: Généralisation à d'autres images:



Paramètres du réseau		Image Initiale (1)	Image résultat (2)
α	= 10 000		
fact_dim	= 0.99		
seuil	= 1		
Nombre d'itération	= 26750		
Nombre d'échantillons	= 600		

Figure 4.11 : Résultats de l'application Cas c.

3-2-1- Interprétation des Résultats:

La base d'apprentissage de ce réseau est constituée d'une image, à partir de laquelle, on extrait des échantillons de taille 3x3. L'opération de l'apprentissage consiste à évaluer les différences entre un pixel central et son voisinage à 8. Cette mesure permet de sélectionner le neurone de la couche compétitive C présentant un poids maximal.

On constate que le nombre d'échantillons appris est fonction du nombre d'itérations. Lors de la phase de la reconnaissance, la taille de l'image à traiter est limitée à 100x100, à cause de l'espace mémoire requis par le programme.

Cas a: On extrait de l'image (1) (Figure 4.9), 9 fenêtres de taille (3x3), l'opération d'apprentissage est terminée au bout de 100 itérations. On applique par la suite l'algorithme de reconnaissance sur l'image ayant servi de base pour l'apprentissage, l'image résultat obtenue présente quelques points dispersés. Certains d'entre eux appartiennent à des contours.

Cas b: On augmente le nombre d'échantillons ainsi que le nombre d'itérations, lors de la phase d'apprentissage. On applique l'algorithme de reconnaissance sur la même image ayant servi de base pour l'apprentissage, On obtient une image où plusieurs frontières délimitant des zones sont mises en évidence, ces frontières sont d'une épaisseur non négligeables, donnant parfois un aspect flou à l'image (Figure 4.10).

Cas c: On applique l'algorithme de reconnaissance sur une autre image n'ayant pas servi de base d'apprentissage. Le résultat obtenu est sous forme de groupement de pixels homogènes et ou de points isolés, correspondant peu à l'image initiale (Figure 4.11).

Conclusion: Le Réseau de Kohonen tel qu'il a été paramétré dans notre application, a donné globalement, des résultats satisfaisants en segmentation. Le réseau a pu apprendre 600 échantillons de taille 3x3. Ainsi, la reconnaissance d'une image ayant servi de base d'apprentissage (i.e. à partir de laquelle, nous avons extrait les échantillons) a donné de bons résultats. Cependant, la généralisation à d'autres exemples non appris, a donné des résultats peu satisfaisants. Aussi, la phase d'apprentissage, nécessite un temps d'exécution très élevé et la taille des images est limitée à 100x100.

CHAPITRE 5:

PRÉSENTATION DU LOGICIEL ET DISCUSSION DES RÉSULTATS

- 1- Environnement de l'expérimentation.
- 2- Présentation du logiciel développé.
- 3- Discussion des résultats.

Chapitre 5: PRÉSENTATION DU LOGICIEL ET DISCUSSION DES RÉSULTATS

1- Environnement de l'expérimentation:

1-1- Configuration de la machine:

L'expérimentation des différents algorithmes a été réalisée sur un micro-ordinateur compatible AT, constitué des éléments suivants:

- Un microprocesseur 586 AMD, de fréquence 133 MHz.
- Une RAM de capacité 08 Mo.

1-2- Langage de programmation:

Le langage de programmation utilisé est le turbo Pascal version 6. C'est un langage évolué et structuré, il permet l'interaction de plusieurs programmes indépendants par la clause UNIT. Il permet aussi l'utilisation de la mémoire en allocation dynamique.

1-3- Description des données de l'expérimentation:

Toutes les applications ont été réalisées sur des images satellitaires prises par le canal XS1 du satellite SPOT, présentant une résolution de 20x20m et représentant un paysage hétérogène de la région de l'Atlas Blidéen, variant du relief montagneux aux zones urbaines et à la couverture agricole.

L'information radiométrique utilisée est sous forme d'une valeur de gris de l'échelle [0,255].

1-4- Structures de données:

Dans l'implémentation des différentes procédures, nous avons utilisé plusieurs types de structures de données.

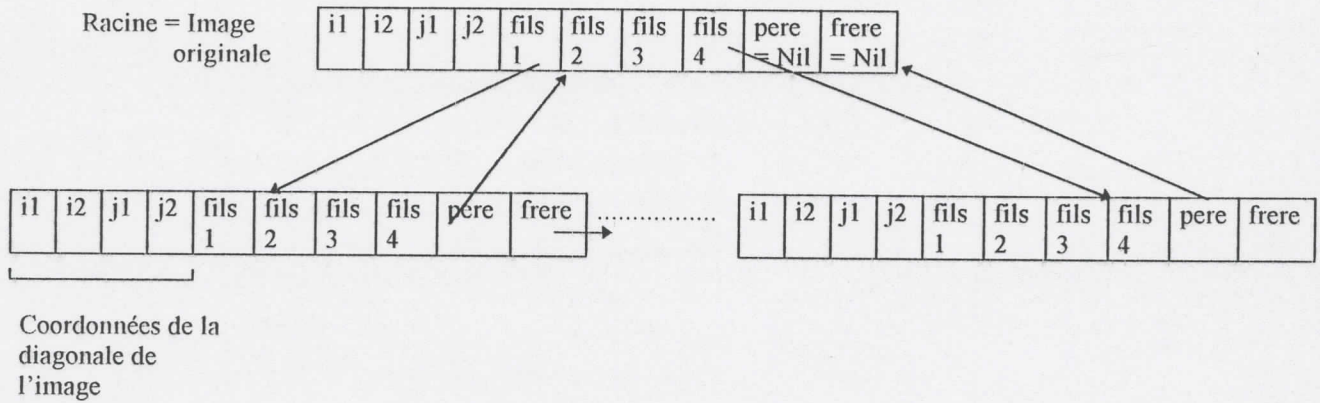
- Fichiers de type byte: Les images sont stockées sous forme de fichiers de bytes. Un byte représente une valeur de niveau de gris.
- Matrices:
 - $A_n(L,K)$: matrices des niveaux de gris aux différentes étapes de traitement.
 - $F(3,3)$: masques des techniques de filtrage.

W...: matrices des poids des connexions pour les R.N.A.

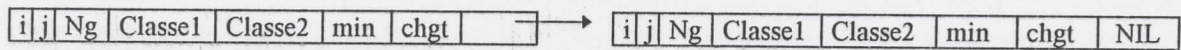
$C_n(L,K)$: les couches des RNA.

- Vecteurs: histogramme des niveaux de gris, densité de la fonction image.
- Variables simples, pour le stockage des différents résultats intermédiaires.
- Liens dynamiques:

1- Structure d'arbre, pour les procédures de partage de l'image en régions homogènes. Chaque noeud est une zone dans l'image:



2- Listes chaînées pour la procédure de segmentation par prétopologie mathématique:



- i,j : coordonnées du pixel.
- NG: niveau de gris du pixel (i,j)
- Classei: Niveaux de gris caractérisant les classes 1 et 2.
- Min: valeur logique indiquant si le pixel (i,j) est un minimum.
- Chgt: valeur logique indiquant si le pixel (i,j) a été visité lors de la dernière itération.

2- Présentation du Logiciel Développé:

Le logiciel conçu est composé de plusieurs unités de programmes, chacune regroupant un ensemble de procédures dédiées à un traitement déterminé:

2-1- Description des unités:

<i>Nom de l'unité</i>	<i>Rôle</i>
INSTALL.bat	<ul style="list-style-type: none"> • Installation du logiciel sur disque dur.
U_PRINC.tpu	<ul style="list-style-type: none"> • Lancement du menu principal.
U_DECL.tpu	<ul style="list-style-type: none"> • Déclarations des objets du logiciel (variables- Types- Constantes).
U_PROC.tpu	<ul style="list-style-type: none"> • Déclarations des procédures communes à toutes les unités: <ol style="list-style-type: none"> 1. Lecture des fichiers images à traiter, 2. Affichage des résultats et 3. Sauvegarde du fichier image résultat.
U_PRET.tpu	<ul style="list-style-type: none"> • Exécution des opérations de prétraitement: <ol style="list-style-type: none"> 1. Transformations ponctuelles. 2. Transformations localisées.
U_SEG.tpu	<ul style="list-style-type: none"> • Exécution des opérations de segmentation relatives à: <ol style="list-style-type: none"> 1. La détection des contours par filtrages différentiel et adaptatif. 2. Seuillage par les algorithmes Isodata et l'Analyse discriminante.
U_REGION.tpu	<ul style="list-style-type: none"> • Exécution de l'opération de segmentation par régions.
U_ALGEBR.tpu	<ul style="list-style-type: none"> • Exécution de l'opération de segmentation par : <ol style="list-style-type: none"> 1. Les opérateurs morphologiques. 2. La prétopologie mathématique.
U_RECUIT.tpu	<ul style="list-style-type: none"> • Exécution des procédures de segmentation basées sur le recuit simulé (RS): <ol style="list-style-type: none"> 1. Recuit 1: Transformations de l'histogramme. 2. Recuit 2: Transformations de l'image 3. Recuit 3: Champs de Markov.
U_RNA.tpu	<ul style="list-style-type: none"> • Exécution des procédures d'apprentissage et de reconnaissance du modèle de Kohonen:

Le schéma suivant présente l'organisation hiérarchique du logiciel:

2-2- Organigramme du logiciel:

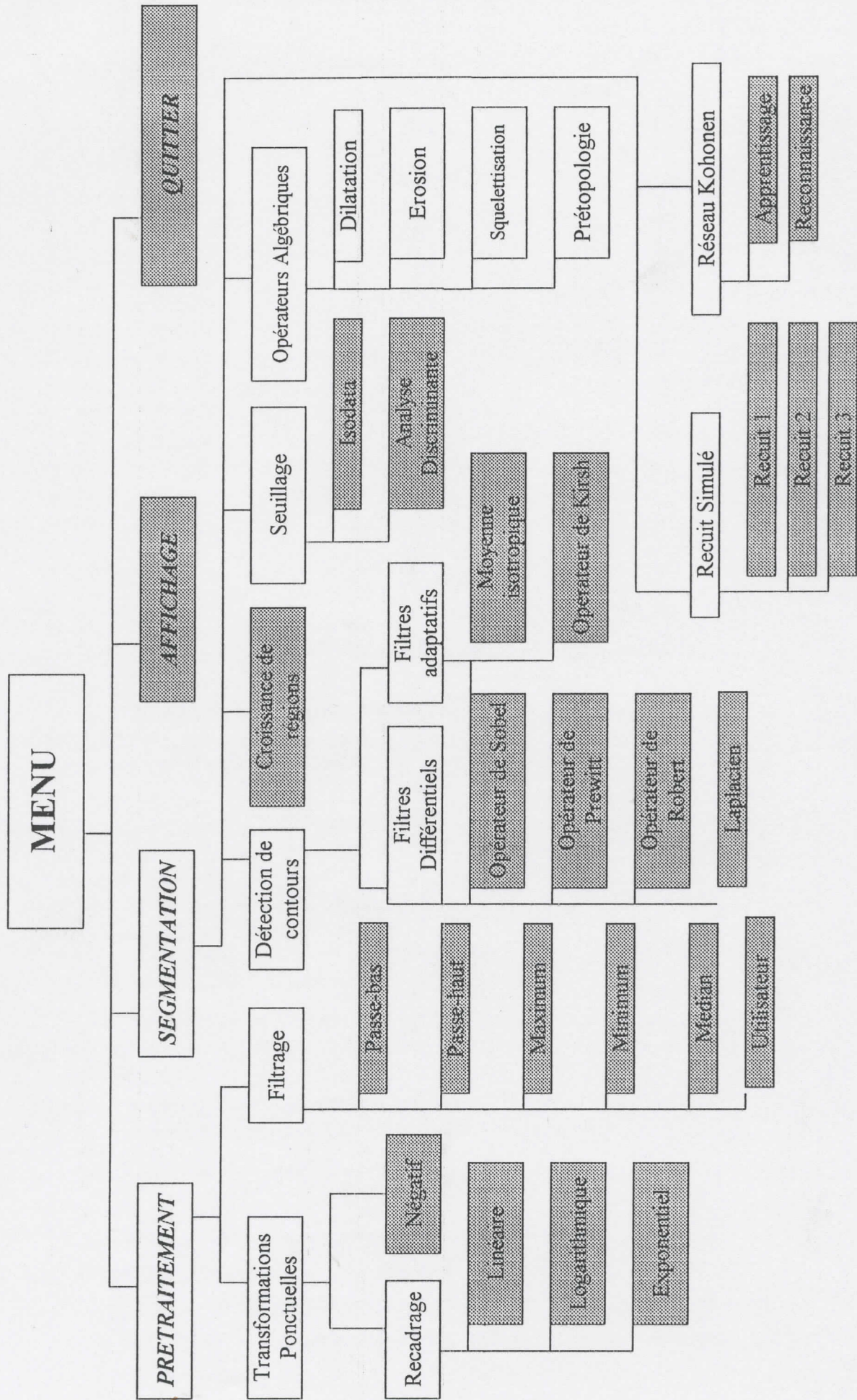


Figure 5.1: Organigramme du logiciel.

Au lancement du logiciel, on voit apparaître le menu principal composé des options suivantes:

PRETRAITEMENT SEGMENTATION AFFICHAGE FIN

Chacune de ces options est activée par sa première lettre. Le choix d'une option quelconque affiche d'autres menus jusqu'à atteindre les opérations de traitement de l'image proprement dites (cases tramées de la figure 5.1).

Toutes ces opérations sont activées par le caractère de couleur rouge (sur écran) (correspondant au caractère majuscule italique dans ce texte) et s'exécutent en quatre étapes:

1- La lecture du fichier image à traiter. Lors de la lecture, l'utilisateur doit introduire, tout d'abord:

- Le chemin du fichier image,
- les coordonnées (x,y) du point origine correspondant au coin gauche supérieur de l'image,
- le nombre de lignes et le nombre de colonnes de l'image à traiter.

Le paramétrage du point origine ainsi que du nombre de colonnes et du nombre de lignes, permet de traiter des fenêtres de l'image.

2- La transformation de l'image.

3- L'affichage des résultats:

- Histogramme de l'image.
- Image graphique.
- Matrice image.

4- La sauvegarde de l'image résultat: Dans ce cas l'utilisateur doit préciser le chemin du fichier image résultat.

2-3- Les opérations de Prétraitement:

2-3-1- Les Transformations Ponctuelles:

- a- *Négatif de l'image*: transforme l'image selon l'équation (1.2).
- b- *Recadrage Linéaire*: transforme l'image selon l'équation (1.3).
- c- *Recadrage logarithmique*: transforme l'image selon l'équation (1.4).
- d- *Recadrage Exponentiel*: transforme l'image selon l'équation (1.5).

2-3-2- Le Filtrage numérique:

Cette option accède aux différents filtres de prétraitement: le filtre passe-Bas, le filtre passe-Haut, le filtre Médian, le filtre minimum, le filtre maximum et le filtre Utilisateur où le masque de convolution est choisi au grès de l'utilisateur.

2-4- Les Opérations de Segmentation:

2-4-1- Détection de contours:

a- Filtrage Différentiel:

Cette option accède aux différents masques de segmentation à caractère différentiel: le filtre de Sobel, le filtre de Prewitt, le filtre de Robert et l'opérateur Laplacien.

b- Filtrage Adaptatif:

Cette option accède à deux filtres adaptatifs qui sont: l'opérateur de Kirsh et l'opérateur Moyenne isotropique.

2-4-2- Seuillage:

Cette option accède à deux procédures de segmentation par seuillage: La procédure Isodata et la procédure Analyse discriminante.

2-4-3- Segmentation par Régions:

Le choix du caractère 'R', déclenche la procédure de segmentation par régions.

2-4-4- Opérateurs Algébriques:

Le choix du caractère 'A', déclenche les procédures algébriques: La Dilatation, l'Erosion, la squelettisation et la Prétopologie mathématique.

2-4-5- Recuit simulé:

Cette option accède à trois procédures de segmentation basées sur le recuit simulé:

- Recuit 1 (Algorithme 3.5).
- Recuit 2 (Algorithme 3.9).
- Recuit 3 (Algorithme 3.11).

2-4-6- Réseau de Kohonen:

Cette option accède à deux procédures basées sur le modèle de Kohonen:

- Apprentissage (Algorithme 4.2).
- Reconnaissance (Algorithme 4.3).

3- Discussion des résultats:

3-1- Conditions de l'expérimentation:

La première critique que l'on peut faire est que le logiciel conçu est un produit prototype. Plusieurs restrictions ont été faites en matière de mémoire vive, ou plutôt dans l'exploitation de la mémoire disponible sur la machine. Les différentes applications ont été réalisées sur des images ne dépassant pas les 170x170 pixels. Plusieurs fois le programme se plante dès qu'on atteint une taille de 200x200 pixels.

3-2- Analyse des espaces mémoires requis:

La figure (5.2), représente la variation de l'espace mémoire requis par les différentes méthodes de segmentation.

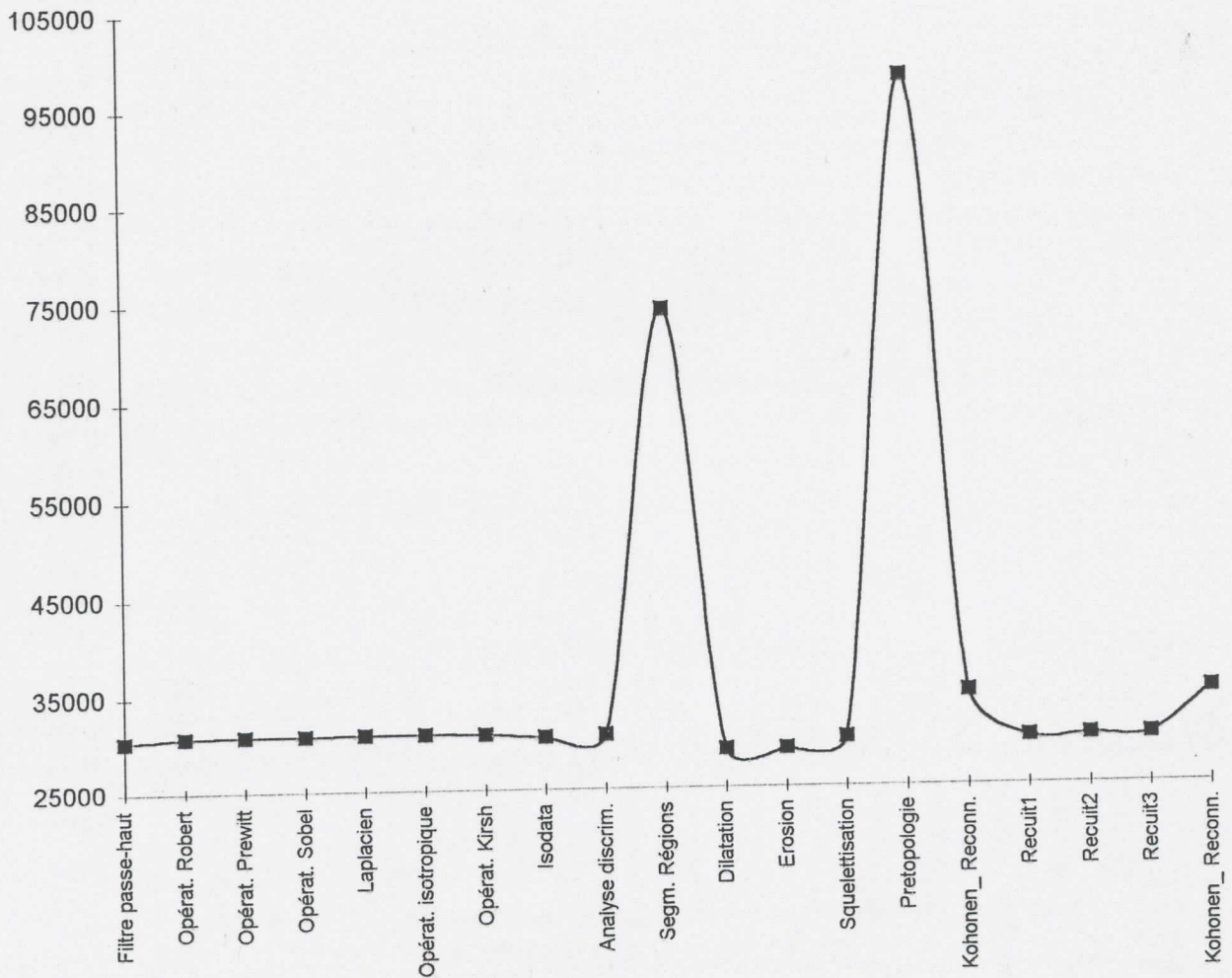


Figure 5.2 : Variation de l'espace mémoire (octets) requis par les méthodes de segmentation.

La figure (5.2), montre que les méthodes de segmentation par prétopologie mathématique ainsi que la segmentation par régions requièrent un espace mémoire relativement élevé par rapport aux autres méthodes. Ceci est dû à l'implémentation de ces deux méthodes qui fait recours à une gestion dynamique de la mémoire.

3-3- Analyse des temps d'exécution:

Les temps d'exécution des différentes techniques de segmentation, sont représentés par la figure (5.3):

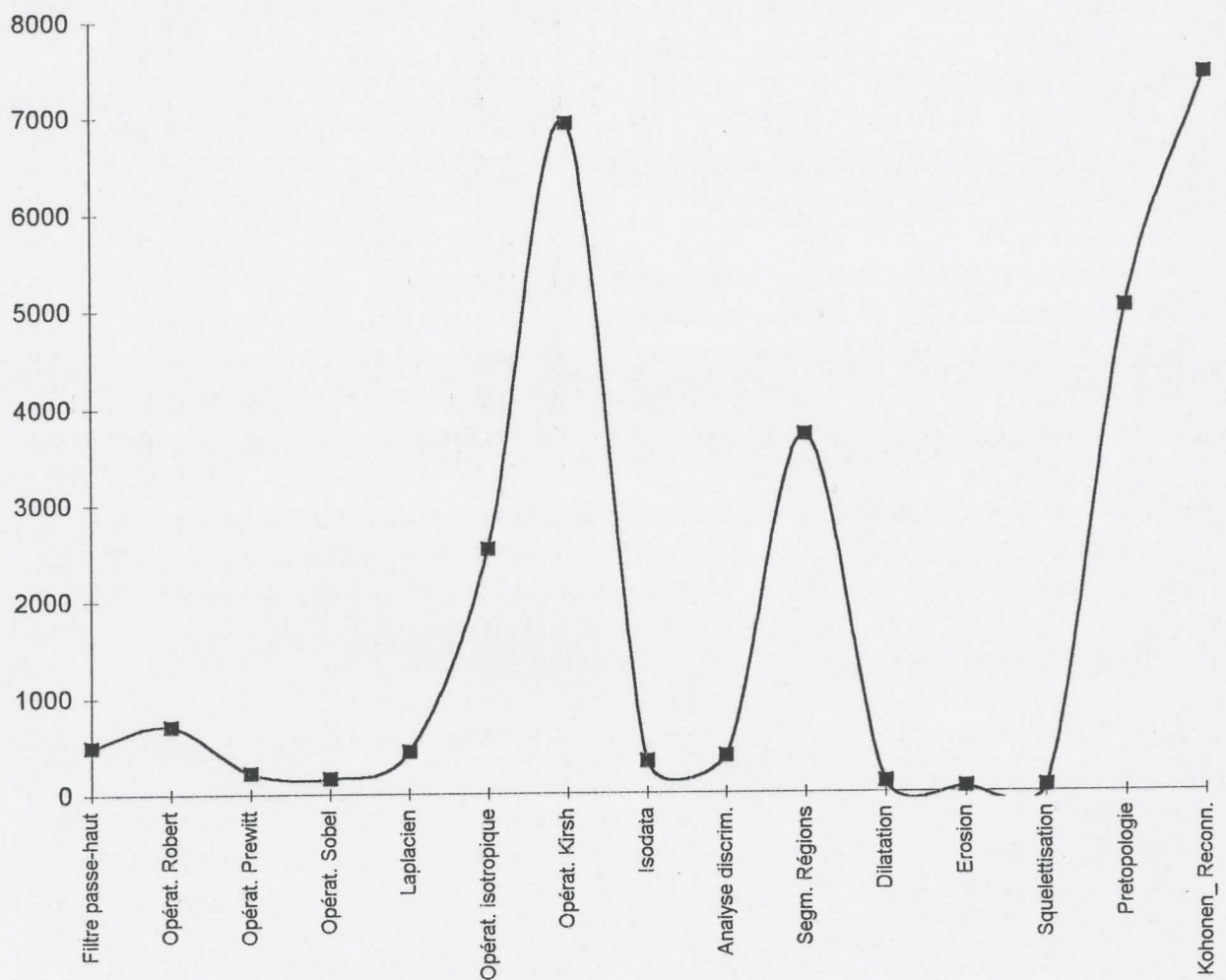


Figure 5.3 : Variation des temps d'exécution (millisecondes) des méthodes de segmentation.

Les filtres adaptatifs présentent le temps d'exécution le plus élevé, notamment l'opérateur de Kirsh. La segmentation par régions et par prétopologie mathématique, présente aussi un temps élevé, car elles sont toutes les deux basées sur un calcul itératif. La

segmentation par régions est une méthode récursive et la deuxième est basée sur un calcul d'adhérence. Les opérateurs morphologiques présentent des temps d'exécution extrêmement faibles. Ceci ne peut être un critère sélectif, car ces méthodes ont été appliquées sur des images binaires seulement.

La figure (5.3) montre que l'algorithme de reconnaissance de Kohonen présente un temps d'exécution élevé. Ceci est dû en fait, à l'opération répétée de l'accès au disque, afin de lire les valeurs des poids des connexions qui sont sauvegardées dans un fichier sur le disque.

Les temps d'exécution des algorithmes du recuit simulé ne sont pas représentés sur la figure (5.3), vu qu'ils sont estimés en plusieurs minutes et parfois en heures.

3-4- Analyse technique des résultats:

L'application des masques de Robert, Prewitt, Sobel, et moyenne isotropique, permet l'extraction de contours épais de l'image. Ces contours sont plus nets sur les images résultant des masques de Prewitt, Sobel, et moyenne isotropique, que sur l'image obtenue par le masque de Robert. Cela est dû au fait que celui-ci opère seulement dans les deux directions diagonales.

Les filtres passe-haut et le laplacien présentent une très bonne extraction de contours dans l'image. Les contours sont parfaitement amincis, mais présentent des discontinuités ne correspondant pas à la réalité. Ces discontinuités sont dues à la sensibilité de ces deux filtres au bruit.

L'algorithme Isodata est déterministe. Il est basé sur le paramètre moyenne des niveaux de gris de l'image. Les résultats obtenus par cet algorithme, montrent que plusieurs zones homogènes sur l'image traitée, sont mises en évidence.

L'algorithme de l'analyse discriminante est basé sur un calcul de variances des niveaux de gris, qui est une mesure de dispersion de ces valeurs autour d'une valeur moyenne. Ceci a permis d'obtenir de meilleurs résultats que ceux obtenus par Isodata, dans le sens où l'on a pu mettre en évidence un plus grand nombre de zones homogènes.

Aussi, ces deux algorithmes de seuillage, ont manifesté une sensibilité à l'égard de la taille de l'image, notamment l'algorithme Isodata. Les résultats obtenus par seuillage local sur différents blocs de l'image sont meilleurs que ceux obtenus par le seuillage global.

Les résultats obtenus par la segmentation par régions, présentent des dilatations locales des zones de niveaux de gris homogènes. Ceci est dû au fait que les images satellites sur lesquelles nous avons réalisé l'expérimentation présentent des variances locales très élevées, par rapport à la variance globale de toute l'image.

L'application des méthodes morphologiques sur une image binaire, améliore l'apparence de l'image, en éliminant les pixels noirs et les pixels blancs isolés respectivement par la dilatation et l'érosion. La squelettisation permet l'extraction de tous les contours de l'image binaire.

La segmentation par prétopologie mathématique est basée sur la propagation des régions et la sélection des pixels appartenant à l'intersection de deux régions adjacentes. Elle permet la détection des frontières des régions.

La segmentation par le recuit simulé est basée sur un calcul de probabilités, donc sur le hasard afin de déterminer le minimum d'une fonction objective. Ceci peut être effectué plus rapidement que les méthodes déterministes, surtout pour les problèmes de grande complexité combinatoire. Il reste que cette approche, pour la même cause du hasard, risque de présenter un nombre d'itérations élevé avant d'atteindre le minimum.

Notre application du recuit simulé au problème de segmentation des images n'a pas donné de bons résultats, si nous les comparons avec les résultats obtenus par les méthodes présentées au chapitre 2. Cependant, les résultats obtenus par les algorithmes Recuit1 et Recuit3, montrent que cette approche peut contribuer à la résolution des problèmes combinatoires se rapportant au traitement des images. En effet, l'algorithme Recuit 1, qui consiste à rechercher la configuration optimale de l'histogramme de l'image et plus particulièrement, l'algorithme Recuit3 qui est basé sur un traitement local de l'image par le biais des champs aléatoires de Markov, ont pu mettre en évidence des zones homogènes sur l'image, mais au prix d'un temps d'exécution plus élevé que les autres méthodes de segmentation.

Les résultats obtenus dans l'application du modèle connexioniste de Kohonen en segmentation, sont tels que, le réseau reconnaît bien une image à partir de laquelle, nous avons extrait les échantillons de l'apprentissage. Mais la reconnaissance d'une nouvelle image n'est pas toujours évidente. Cela permet de dire que la généralisation pour des cas (images) non appris n'est possible que si ces images sont constituées des fenêtres extraites de l'image ayant servi de base d'apprentissage.

3-5- Récapitulation de la discussion:

Le tableau suivant résume les principaux avantages et inconvénients que nous avons pu déduire au terme de cette étude:

Méthodes	Avantages	Inconvénients	Autres caractéristiques
Filtres: Robert, Sobel, Prewitt, Moy. isotropique	<ul style="list-style-type: none"> - Implémentation simple. - Gestion statique de la mémoire. - Détection des contours épais. 	<ul style="list-style-type: none"> - Sensibilité au bruit. 	<ul style="list-style-type: none"> - Déterministe.
Filtre passe-haut et le laplacien.	<ul style="list-style-type: none"> - Implémentation simple. - Gestion statique de la mémoire. - Détection des contours amincis. 	<ul style="list-style-type: none"> - Sensibilité au bruit. 	<ul style="list-style-type: none"> - Déterministe
Opérateur de Kirsh	<ul style="list-style-type: none"> - Implémentation simple. - Gestion statique de la mémoire. 	<ul style="list-style-type: none"> - Image résultat floue. - Temps d'exécution élevé. 	<ul style="list-style-type: none"> - Déterministe.
Méthodes de seuillage.	<ul style="list-style-type: none"> - Implémentation simple. - Gestion statique de la mémoire. - Mise en évidence de zones homogènes. 	<ul style="list-style-type: none"> - Sensibilité à l'égard de la taille de l'image traitée. 	<ul style="list-style-type: none"> - Déterministe. - Itérative.
Segmentation par Régions.	<ul style="list-style-type: none"> - Dilatation des zones claires. 	<ul style="list-style-type: none"> - Temps d'exécution élevé. - Gestion dynamique de la mémoire. - Taille limitée de l'image. 	<ul style="list-style-type: none"> - Déterministe. - Itérative.
Prétopologie mathématique	<ul style="list-style-type: none"> - Mise en évidence de contours de l'image. 	<ul style="list-style-type: none"> - Implémentation compliquée. - Temps d'exécution élevé. - Gestion dynamique de la mémoire. - Taille limitée de l'image. 	<ul style="list-style-type: none"> - Déterministe. - Itérative.
Opérateurs morphologiques	<ul style="list-style-type: none"> - Implémentation simple. - Gestion statique de la mémoire. - Temps d'exécution faible - Détection des contours par la squelettisation. 	<ul style="list-style-type: none"> - Application aux images binaires seulement dans ce cas, 	<ul style="list-style-type: none"> - Déterministe.
Recuit Simulé	<ul style="list-style-type: none"> - Gestion statique de la mémoire. - Mise en évidence de certaines zones homogènes par Recuit1 et Recuit3 	<ul style="list-style-type: none"> - Temps d'exécution élevé. - Résultats aberrants par Recuit2. 	<ul style="list-style-type: none"> - Stochastique.
Réseau de Kohonen	<ul style="list-style-type: none"> - Gestion statique de la mémoire. - Reconnaissance de cas appris. 	<ul style="list-style-type: none"> - Taille de l'image limitée. - Généralisation aux cas non appris n'est pas évidente. 	<ul style="list-style-type: none"> - Estimation subjective de l'apprentissage.

Tableau 5.1: Avantages et Inconvénients des méthodes de segmentation.

CONCLUSION GENERALE

CONCLUSION GENERALE

La segmentation des images est une étape cruciale en traitement d'images. Elle ne cesse de mobiliser autour d'elle beaucoup de chercheurs en quête d'améliorer les résultats obtenus jusqu'alors. Le travail que nous avons réalisé est une modeste contribution à la compréhension de ce phénomène. Ainsi, nous avons, dans le cadre de cette thèse, présenté une étude comparative sur un ensemble non exhaustif de méthodes de segmentation d'images que nous avons appliqué sur des images satellites Spot. Le choix de ce type d'application, relève d'un intérêt particulier que nous portons au domaine de la télédétection. Aussi, l'indisponibilité des autres types d'images nous a beaucoup aidé à opter pour ce choix.

L'étude comparative a concerné quatre classes de méthodes de segmentation « classiques »: (filtrage, seuillage, segmentation par régions et méthodes algébriques), ainsi que l'approche aléatoire du recuit simulé et l'approche connexioniste de Kohonen que nous avons appliquées, au problème de segmentation des images à niveau de gris.

Les méthodes de filtrage ont fourni de très bons résultats, notamment les filtres de Sobel, Prewitt, l'opérateur isotropique et le laplacien.

Les méthodes de seuillage, ont aussi fourni de bons résultats en segmentation, sauf que celle-ci est meilleure si le seuillage de l'image est effectué localement sur des blocs de l'image.

La segmentation par régions homogènes, des images spatiales, contribue à une dilatation des zones de niveau de gris extrême, elle ne peut être retenue comme moyen de segmentation de ce type d'images.

L'application des méthodes morphologiques sur une image binaire, améliore l'apparence de l'image, en éliminant les pixels noirs et les pixels blancs isolés respectivement par la dilatation et l'érosion. La squelettisation permet l'extraction de tous les contours de l'image binaire. La prétopologie, par son principe de propagation dans la recherche des zones minimales, permet aussi une extraction des contours de l'image. Nous avons combiné à cette méthode, un moyen de sélection des contours les plus significatifs.

Dans le troisième chapitre, nous avons utilisé l'approche aléatoire du recuit simulé, en segmentation des images. Par cette application, nous cherchions à valider l'efficacité du recuit simulé dans des cas pratiques d'optimisation de problèmes combinatoires. D'une manière globale, les résultats obtenus ne sont pas aussi bons que ceux obtenus par les autres méthodes. Il reste que l'algorithme Recuit1 et plus particulièrement l'algorithme Recuit3 qui fait intervenir les champs Markoviens ont donné, au prix d'un temps de réponse élevé, des résultats prometteurs et perfectibles: l'utilisation d'autres fonctions objectives pourrait donner de meilleurs résultats.

Dans le chapitre quatre, nous avons aussi utilisé le modèle de réseaux de neurones de Kohonen, à des fins de segmentation. L'algorithme basé sur la grille bidimensionnelle de Kohonen, utilise une seule couche compétitive. Les entrées sont codées par la somme des écarts des niveaux de gris d'un même voisinage. Cet algorithme a pu extraire certains contours de l'image. Mais, il présente l'inconvénient de ne pouvoir traiter que des images de petite taille 100x100. Aussi, la généralisation à des cas non appris, n'a pas donné de bons résultats.

Enfin, après avoir essayé de modéliser la segmentation des images selon l'approche aléatoire du recuit simulé et l'approche connexioniste de Kohonen, nous constatons qu'il reste encore beaucoup à faire, nous avons l'intention de paralléliser le phénomène de recherche des minimums de la fonction objective par Recuit 3, en le modélisant par un réseau de neurones. Cela, pourrait peut être améliorer la qualité des résultats et diminuer le temps de réponse.

Aussi, à la suite de l'étude comparative des méthodes de segmentation appliquées à des images satellites et étant donné que ces images constituent une représentation graphique d'un milieu naturel complexe. Nous concluons que chaque technique présente des avantages et des inconvénients et permet donc la résolution d'un problème et non de tous les problèmes, la solution voulue doit être étudiée de cas en cas. Nous pourrions ici, rajouter un appui de N.Benhadjel[7] « *Il reste très difficile de répondre à la question: qu'est ce qu'un bon algorithme de segmentation? Mais le meilleur moyen de juger la performance d'une segmentation de scènes très complexes reste encore une évaluation subjective, par un observateur humain* ».

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] AARTS E.H.L. and KORST J.
« **Simulated Annealing and Boltzman Machines: a Stochastic Approach to combinatorial optimization and Neural computing** »
Wiley 1989.
- [2] AARTS E.H.L. and LAARHOVEN P.J.M.
« **Statistical Cooling: A General Approach to Combinatorial Optimisation Problems** »
Journal of Research 40, 1985.
- [3] ABDELALI N.
« **Recherche d'une trajectoire sans collision pour un robot mobile** »
Thèse de Magister, Institut d'Informatique, USTHB, 1993.
- [4] AIT AKKACHE M.
« **Les Réseaux de neurones séquentiels, application aux chaînes de Markov** »
Thèse de Magister, Institut de Mathématiques appliquées, USTBlida, 1996.
- [5] BECKER F.
« **Bases de la télédétection et sa problématique** »
Revue la Houille blanche / N° 7/8 1978
- [6] BELBACHIR A.H. , BACHARI N.I. et BENABADJI N.
« **Numerical Methods for Satellite Imagery Analysis** »
AMSE Journal , volume 38 N°1,2/1997.
- [7] BENHADJEL BOUDJEMAA N.
« **Modélisation Floue de l'Incertitude pour la Segmentation d'Images** »
Thèse de Doctorat, Univ. Rene Descartes Paris V, 1993
- [8] BESAG J.
« **Spatial Interaction and the statistical analysis of lattice systems** »
Journal of Royal Statistical Society #2, pp. 192-236, 1974
- [9] BONOMI E. and LUTTON J.L.
« **The N-city Travelling Salesman Problem: Stastical Mechanics and The Metropolis Algorithm** »
SIAM Rev. 26 , 1984.
- [10] BONOMI E. and LUTTON J.L.
« **Le recuit simulé** »
Science, N°129, 1988.
- [11] BOURRET P., REGGIA J. et SAMUELIDES M.
« **Les Réseaux neuronaux , une application connexioniste de l'Intelligence Artificielle** »
TEKNEA, 1991.

- [12] CANNY J.F.
« **A Computational Approach to Edge Detection** »
IEEE Trans. on PAMI, Vol 8, N°6, 1986.
- [13] CERVELLE B. et CHOROWICZ J.
« **La télédétection des ressources minérales** »
Revue la RECHERCHE , N°174, Fev 1986.
- [14] COQUERET J.P. et PHILIPP S.
« **Analyse d'images: filtrage et segmentation** »
Ed. MASSON, 1995.
- [15] DECARVALHO GOMES F. A.
« **Utilisation d'algorithmes stochastiques en apprentissage** »
Thèse de Doctorat. Université des Sciences et Techniques du Languedoc. Montpellier II, 1992.
- [16] Djar F.
«**Résolution du problème du voyageur de commerce par la méthode du recuit simulé**»
Mémoire de fin d'études, Inst. des sciences exactes, Dépt. Mathématiques appliquées,
USTBlida, 1993.
- [17] EMPTOZ H.
« **Modèle prétopologique pour la reconnaissance de formes. Application en neurophysiologie** »
Thèse de Doctorat , Univ. Claude Bernard Lyon I, 1983.
- [18] FETTOUS N.
« **La méthode du recuit simulé et ses applications** »
Mémoire de fin d'études, Inst. des sciences exactes, Dépt. Mathématiques appliquées,
USTBlida, 1996.
- [19] GAUCHER P.
« **Segmentation d'images numériques par mesure de proximité de deux pixels adjacents** »
Thèse de Doctorat , Université de Tours, 1992.
- [20] GEMAN D. et GEMAN S.
« **Boundary Detection by Constrained Optimization** »
IEEE Trans. on PAMI , Vol 12, 609-628, 1990.
- [21] GOLDBERG M.
« **Processing of Remote Sensing Data and Optimization**»
University of Ottawa, Canada, 1986.
- [22] GUYON X.
« **Algorithmes Stochastiques et Traitement d'Images** »
1ères journées Maghrébines de Mathématiques Appliquées Bizerte-Tunisie
Du 1er au 05 Novembre 1994.

- [23] GUYON X.
« **Random Fields on network: Modeling, Statistics and applications** »
Série: Probabilities and its application. Ed. SPRINGER VERLAG, 1995.
- [24] HARALICK R.M.
« **Ridges and Valleys on Digital Images** »
Computer Graphics an Image Processing, N°22, p 28-38, 1983.
- [25] HEBB S.
« **The Organization of Behavior** »
Wiley and Sons , New York, 1949.
- [26] HOCQUENGHEM A.
« **Elément de calcul différentiel et intégral** »
Série Mathématiques, Tome 2. ED. Masson, 199?
- [27] HUANG M.D. et al
« **An Efficient General Cooling Schedule for Simulated Annealing**»
Proceeding IEEE, International Conference on Computer-Aided Design. Santa Clara, 1986
- [28] JOHNSON D.S. et al
« **Optimisation by Simulated Annealing: An Experimental Evaluation** »
Revue Research 3, 1991
- [29] KIM IL Y. and YANG HYUN S.
« **Efficient Image Labeling based on Markov Random Fields and Error backpropagation Network** »
The journal of Pattern Recognition Society, Vol. 26 N°11, 1993
- [30] KIRKPATRICK S., GELATT C. et VECCHI M.
« **Optimization by Simulated Annealing** »
IBM Research Report 9353, 1982.
- [31] KIRKPATRICK S., GELATT C. et VECCHI M.
« **Optimization by Simulated Annealing** »
Science 4598. Vol 220, 1983.
- [32] KOHONEN T.
« **Self Organization and Associative Memory** »
Ed. Springer Verlag ,1988.
- [33] KUNNT ET al.
« **Traitement Numérique des Images**»
Lavoissier, 1991.
- [34] LUNDY M. et MEES A.
« **Convergence of an Annealing Algorithm** »
Mathematical Programming, 1986.

- [35] MASSON P.
« **Etude d'algorithmes de classification contextuelle et application à la Segmentation d'images satellite** »
Thèse de Doctorat, Univ. de Bretagne Occidentale, 1991.
- [36] McCULLOCH W. and PITT W.
« **A Logical Calculus of the Ideas Immanent in Nervous Activity** »
Bulletin of Mathematical Biophysics, 1943.
- [37] METROPOLIS N. , ROSENBLUTH A. , TELLER A. et TELLER E.
« **Equation of state calculations by Fast Computing Machines** »
Journal of Chem. Physics, N° 21, 1953.
- [38] MINSKY M. et PAPERT S.
« **Perceptrons** »
MIT Press, Cambridge, MA, 1988.
- [39] MOUSSAOUI A..
« **Identification d'objets à partir d'images satellitaires en utilisant un modèle direct** »
Rapport de Magister en Génie spatiale, Ins. d'Informatique, USTOran, 94.
- [40] MRAGHNI M. C.
« **Détection de Chaînes de Contours d'une Image Numérique par Approche Symbolique et par Grammaire de Formes** »
Thèse de Doctorat, Ecole d'Ing. en Informatique pour l'Industrie, Univ. de Tours, 1997.
- [41] NALWA V. S. et BINFORD T.O.
« **On Detecting Edges** »
IEEE Trans. on PAMI, Vol 8, N°6, 1986.
- [42] PAVLIDIS T.
« **Why Progress In Machine Vision Is So Slow** »
Pattern Recognition Letters, Vol 13, 1992.
- [43] POSTAIRE J.G.
« **De l'image à la décision** »
Edition DUNOD, 1987.
- [44] PRATT W.K.
« **Digital Image Processing** »
Wiley Inter Science Publication, New York, 1978
- [45] ROSENBLATT F.
« **Principles of Neurodynamics** »
Spartan Press, East Lansing, MI, 1962.
- [46] ROUGAB A.R.
« **Mise au point d'un logiciel conversationnel de traitement d'images satellitaires orientés vers les applications urbaines** »
Thèse de Magister, Institut d'Electronique, ENP d'Alger, 1996.

- [47] SECHEN C. et al
« **The Timber Wolf Placement and Routing Package** »
Solid State Circuits. IEEE, 1985
- [48] SELMAOUI N.
« **Les lignes de crêtes dans les images à niveaux de gris. Contribution de la prétopologie et de la classification automatique à leur détection. Taxinomie des méthodes** »
Thèse de Doctorat, Institut des sciences appliquées de Lyon, 1992.
- [49] TAGARE H. D. et DeFIGUEIREDO R. P.
« **On the Localisation Performance Measure and Optimal Edge Detection** »
IEEE Trans. on PAMI, Vol 12, N°12, 1990.
- [50] TAHRAOUI D.
« **Etude expérimentale des performances de quelques modèles de réseaux de neurones appliqués à la reconnaissance des caractères** »
Thèse de Magister, Institut d'Electronique, USTBlida, 1997.
- [51] TORRE V. et POGGIO T. A.
« **On Edge Detection** »
IEEE Trans. on PAMI, Vol 8, N°2, 1986.
- [52] TOUMAZET J. J.
« **Traitement de l'image par l'exemple** »
Ed. SYBEX, 1990.
- [53] WESZKA J.S. ET ROSENFELD A.
« **A threshold Selection Technique** »
IEEE Trans. on Computers Vol 23, 1322-1326, 1974.
- [54] WINKLER G.
« **Image analysis, Random fields and Dynamic Monte Carlo methods** »
Ed. SPRINGER VERLAG, Berlin 1995.
- [55] XU L. and Erkki O.
« **Improved simulated Annealing Boltzman machine and attributed graph matching, Neuronal Networks** »
Ed. L.B. ALNEIDA C.J. WELLEKENES, Portugal 1990.
- [56] ZERUBIA J. et PLOYETTE F.
« **Détection de contours et Lissage d'image par deux algorithmes déterministes de relaxation. Mise en oeuvre sur la machine à connexions CM2** »
Revue Traitement du signal, Volume 8, N°3, pp 165-175, 1991.