

République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT AUTOMATIQUE

MEMOIRE DE MAGISTER
en *Automatique*

Option: **Traitement d'Images et**
Reconnaissance de Formes

Présenté par

ABDELLI OUARDIA

Thème

Segmentation d'images par seuillage d'histogrammes
bidimensionnels.

Mémoire soutenu le : 02/07 /2011

devant le jury d'examen composé de :

DIAF MOUSSA

Professeur à l'U.M.M.T.O.

Président

HAMMOUCHE KAMAL

M.C.A. à l'U.M.M.T.O.

Rapporteur

OUANES MOHAND

M.C.A. à l'U.M.M.T.O.

Examineur

KARA REDOUANE

M.C.A. à l'U.M.M.T.O.

Examineur

Résumé : Les méthodes de seuillage d'histogrammes sont souvent considérées comme des méthodes de segmentation d'images privilégiées. Ces méthodes reposent sur l'exploitation de l'histogramme monodimensionnel de l'image qui caractérise la distribution des niveaux de gris. Cependant les performances de ces méthodes se dégradent rapidement lorsque les images à seuiller sont trop bruitées. Afin d'éviter ce problème des méthodes plus robustes basées sur l'histogramme bidimensionnel ont été proposées. Contrairement à l'histogramme monodimensionnel, construit en tenant compte que du niveau de gris d'un seul pixel, l'histogramme bidimensionnel tient compte à la fois du niveau de gris d'un pixel et de l'information locale. Dans ce cas un couple de seuils doit être déterminé. Cependant cette extension nécessite un temps de calcul très élevé. L'objectif de ce mémoire consiste à accélérer les calculs en faisant appel aux techniques d'optimisation méta-heuristiques.

Avant propos

Le travail que nous présentons dans ce mémoire a été effectué au sein de laboratoire « Robotique et vision » du département automatique, Faculté de Génie Electrique et Informatique, Université Mouloud Mammeri, Tizi-Ouzou.

J'exprime mes remerciements et ma gratitude à mon directeur de mémoire, Monsieur **HAMMOUCHE KAMAL** maître de conférences classe A à l'Université Mouloud Mammeri, pour l'aide et le soutien et les conseils qu'il m'apportés tout au long de ce travail.

Nos vifs remerciements vont aussi à Monsieur **DIAF MOUSSA** professeur à l'Université Mouloud Mammeri, pour avoir fait l'honneur de présider le jury de ce mémoire et l'intérêt qu'il a apporté à ce travail.

Nous tenons à remercier Monsieur **KARA RADOUANE**, maître de conférences classe A à l'Université Mouloud Mammeri, pour avoir bien voulu faire partie de du jury de ce mémoire.

Monsieur **OUANES MOHEND**, maître de conférences classe A à l'Université Mouloud Mammeri à bien voulu participer au jury de ce mémoire, qu'il trouve, ici, l'expression de notre profonde gratitude pour l'intérêt qu'il a apporté à notre travail.

Nous tenons aussi à remercier vivement tout les membres de laboratoire Robotique et vision pour l'aide qu'ils nous ont apportée.

Ce mémoire est dédié à ma famille: mon père, ma mère dont la fierté est ma raison de vivre; mes frères et mon mari Ali et toute sa famille. Rien de tout ceci ne serait possible sans leurs sacrifices et leurs encouragements.

Sommaire

SOMMAIRE

Introduction générale	1
Chapitre 1 : Etat de l'art sur la segmentation d'images	
1.1. Introduction.....	3
1.2. Définition de la segmentation d'image.....	3
1.3. Techniques de segmentation.....	4
1.3.1. Approche contour.....	4
1.3.2. Approche région.....	6
1.3.2.1. Croissance des régions.....	6
1.3.2.2. Division des régions	7
1.3.2.3. Division/fusion.....	7
1.3.2.4. Segmentation d'image par classification	8
1.3.3. Approche coopérative.....	9
1.4. Segmentation par seuillage	10
1.4.1. Définition du seuillage	10
1.4.2. Méthodes de seuillage globale.....	11
1.4.2.1. Méthodes paramétriques.....	12
1.4.2.2. Méthodes non paramétriques.....	13
1.4.2.2.1. Méthode d'Otsu.....	13
1.4.2.2.2. Méthodes de Kittler et Illmgworth.....	14
1.4.2.2.3. Méthode de Kapur :	16
1.4.2.2.4. Méthode de cross entropie	16
1.4.2.2.5. Méthode de corrélation entropique.....	17
1.4.2.2.6. Méthode d'entropie de Renyi.....	18
1.4.2.2.7. Méthode d'entropie de Tsallis-Havrda-Charvat	19
1.4.2.2.8. Méthode d'entropie exponentielle.....	20
1.4.3. Méthodes de seuillage locale ou adaptatif.....	21
1.5. Conclusion.....	23

Chapitre 2 : Seuillage d'histogrammes bidimensionnel et tridimensionnel

2.1. Introduction.....	24
2.2. Histogrammes bidimensionnels.....	24
2.2.1. Matrice de cooccurrence.	24
2.2.2. Histogramme bidimensionnel proprement dit.....	25
2.2.3. D'autres histogrammes bidimensionnels	26
2.3. Méthodes de seuillage basées sur les matrices de cooccurrence.....	26
2.4. Seuillage d'histogrammes bidimensionnels	29
2.4.1. Seuillage d'Otsu bidimensionnel.....	30
2.4.2. Entropie de Shannon à deux dimensions	31
2.4.3. Entropie exponentielle bidimensionnelle	32
2.4.4. Entropie de Renyi bidimensionnelle.....	33
2.4.5. Entropie de Tsallis-Havrda-Charvat bidimensionnelle.....	35
2.5. Histogramme à trois dimensions	36
2.5.1. Méthode d'Otsu tridimensionnelle.....	38
2.5.2. Entropie de Kapur à trois dimensions	39
2.5.3. Entropie exponentielle tridimensionnelle.....	40
2.5.4. Entropie de Renyi tridimensionnelle.....	41
2.5.5. Entropie de Tsallis-Havrda-Charvat tridimensionnelle.....	42
2.6. Conclusion	43

Chapitre 3 : Etat de l'art sur les méthodes métaheuristiques

3.1 Introduction	44
3.2 Définitions des notions d'heuristiques et de métaheuristiques	44
3.2.1. Heuristiques et Métaheuristiques.....	44
3.2.2. Notion de voisinage.....	45
3.3. Méthodes d'optimisation métaheuristiques	45
3.3.1. Les métaheuristiques à solution unique.....	45
3.3.1.1. Le recuit simulé (Simulated Annealing)	47
3.3.1.2. Le recuit microcanonique.....	48
3.3.1.3. La recherche Tabou	50
3.3.2. Les métaheuristiques à population de solutions	51

3.3.2.1. Les algorithmes génétiques (Genetic Algorithms)	52
3.3.2.2. Algorithmes de l'évolution différentielle (differential evolution)...	54
3.3.2.3. Les colonies de fourmis (Ants System)	55
3.3.2.4. Systèmes immunitaires artificiels (Artificial Immune Systems) :	57
3.3.2.5. Optimisation par colonie d'abeille.....	58
3.3.2.6. Algorithme par sauts de grenouilles	61
3.3.2.7. L'optimisation par essais particuliers(OEP)	63
3.3.2.7.1. Principe de base d'un PSO.....	65
3.3.2.7.2. L'algorithme d'OEP.....	66
3.3.2.8. Autres métaheuristiques	67
3.4. Conclusion	68

Chapitre 4 : Testes et Résultats

4.1. Introduction	69
4.2. Critères d'évaluation.....	69
4.3. Images synthétiques.....	69
4.3.1. Calcul exhaustif des seuils.....	70
4.3.2. Interprétation des résultats.....	73
4.3.3. Calcul des seuils par la méthode d'essaim de particules.....	74
4.3.3.1. Paramètres du PSO	74
4.3.3.2. Résultats du PSO.....	74
4.3.3.3. Interprétation des résultats.....	78
4.4. Images réelles.....	79
4.4.1. Paramètres du PSO pour les images réelles.....	80
4.4.2. Résultats du PSO.....	81
4.4.3. Interprétation des résultats	84
4.5. Conclusion.....	85
Conclusion générale	86

Introduction générale

Introduction générale

La segmentation d'images est l'opération la plus importante dans un système de traitement des images, car elle est située à l'articulation entre le traitement et l'analyse des images. L'intérêt de la segmentation est de partitionner une image en plusieurs régions homogènes, au sens d'un critère fixé a priori. L'intérêt de disposer de régions homogènes est de fournir des données simplifiées, qui facilitent la tâche d'un système de reconnaissance de formes, ou autre système d'extractions des objets contenus dans l'image.

Le seuillage (thresholding en Anglais) représente un outil largement utilisé dans la segmentation d'images pour extraire des objets de leurs fonds en fonction d'un seuil. Tout problème de seuillage consiste alors à rechercher la valeur du seuil. De par sa simplicité, il occupe une place prépondérante dans plusieurs applications telles : l'analyse de documents, analyse d'images biomédicales etc. La plus part des méthodes de seuillage déterminent le seuil en optimisant une fonction objective, elles sont basées sur l'histogramme monodimensionnel de l'image. Cependant, ces méthodes ne tiennent pas en compte des relations spatiales entre les pixels et peuvent être dans certaines situations (bruit, mauvais contraste) peu performantes.

Afin d'améliorer les résultats de segmentation, des méthodes basées sur des histogrammes bi et tridimensionnels ont été alors proposées dans la littérature. Ces méthodes ne sont que des extensions simples des méthodes monodimensionnelles au cas multidimensionnelles. Elles consistent à chercher un couple de seuils dans le cas des histogrammes bidimensionnels et un triplet de seuils dans le cas tridimensionnels.

Les techniques de seuillage des histogrammes ont été également généralisées dans le cas du multi-seuillage (multi-level thresholding en Anglais). Le multi-seuillage est utile quand on a affaire à des images qui contiennent plusieurs objets ayant des luminances différentes. Pour extraire ces objets, plusieurs seuils sont nécessaires.

Nous proposons dans ce mémoire d'étudier certaines méthodes de seuillage d'histogrammes bidimensionnels, de les appliquer dans la recherche de plusieurs seuils et de les étendre au cas d'histogrammes tridimensionnels. Cependant, toutes ces extensions nécessitent un temps de calcul très élevé. Pour résoudre ce problème, nous proposons dans ce

mémoire de déterminer les seuils en utilisant la méthode d'optimisation métaheuristique par essaim de particules (PSO).

Nous avons scindé notre travail en quatre chapitres : Le premier chapitre donne quelques notions de segmentation et présente quelques méthodes de seuillage globale.

Le chapitre 2 est consacré à la présentation d'une panoplie de méthodes de seuillage bidimensionnel et leurs extensions dans le cas tridimensionnel et dans le cas du multi-seuillage.

Le chapitre 3, est consacré à la présentation d'un état de l'art sur les méthodes d'optimisation métaheuristique.

Le chapitre 4 est réservé à la présentation des tests et des résultats de l'algorithme de PSO dans le seuillage des histogrammes monodimensionnels, bidimensionnels, tridimensionnels. Une conclusion générale clôturera ce mémoire.

Chapitre 1

Etat de l'art sur la segmentation d'images

1.1. Introduction

L'analyse et l'interprétation des images sont actuellement très utilisées dans différents domaines allant de l'aide au diagnostic, en médecine, à la navigation autonome des véhicules en passant par la reconnaissance des visages, des empreintes, de l'iris, le contrôle de qualité des produits manufacturables (bois, tissus, verre, pièces mécaniques)..., etc. Dans un système d'analyse d'images, la segmentation apparaît comme l'étape la plus importante car toutes les tâches ultérieures comme l'extraction de primitives, la détection d'une position d'un objet, ou la reconnaissance d'un objet dépendent fortement de la qualité de la segmentation. Le but de la segmentation d'image est de faciliter l'extraction des éléments qui la composent. Pour cela, plusieurs approches ont été proposées. Nous présenterons dans ce chapitre une brève revue de ces approches puis nous détaillerons quelques méthodes de seuillage d'histogrammes.

1.2. Définition de la segmentation d'image

La segmentation permet d'isoler dans l'image les objets sur les quels doit porter l'analyse ou de séparer les régions d'intérêt de son fond. Autrement dit, elle permet de partitionner l'image en zones homogènes ayant des caractéristiques (niveau de gris, couleur, texture) identiques où une zone peut correspondre un objet ou une partie d'un objet. Mathématiquement, la segmentation est définie par Zucker de la manière suivante [Zucker, 1976]:

Segmenter une image I en n régions, revient à la partitionner en n sous ensembles R_1, R_2, \dots, R_n tels que :

1. $I = \bigcup_i R_i$.
2. R_i est constituée de pixels connexes pour tout i .
3. $P(R_i) = \text{Vrai}$ pour tout i .
4. $P(R_i \cup R_j) = \text{faux}$ pour tous i, j , R_i et R_j étant adjacentes dans I .

La première condition indique que chaque pixel de l'image doit appartenir à une région R_i et l'union de toutes les régions forme l'image entière. La deuxième condition est relative à la structure des régions, elle définit une région comme un ensemble de pixels qui doivent être connexes. La troisième condition exprime que chaque région doit respecter un prédicat d'uniformité. La dernière condition implique la non réalisation de ce même prédicat pour la réunion de deux régions adjacentes.

L'opération de segmentation est souvent confrontée aux problèmes d'ambiguïtés et du bruit qui affectent certains pixels de l'image. C'est la raison pour laquelle, plusieurs techniques de segmentation ont été proposées.

1.3. Techniques de segmentation

On peut distinguer trois grandes approches de segmentation : contour, région et hybride.

1.3.1. Approche contour

Il s'agit dans cette approche de rechercher les contours qui délimitent les régions homogènes de l'image. Un contour est un ensemble de pixels formant une frontière entre deux ou plusieurs régions voisines, il est défini par une variation "rapide" du caractéristique niveau de gris, couleur ou texture. La recherche des contours dans une image a été étudiée depuis l'origine des travaux sur l'imagerie numérique.

Il existe plusieurs méthodes de segmentation basées sur l'approche contour qu'on peut regrouper en trois catégories: les méthodes dérivatives, par filtrage optimal et les contours actifs.

Les méthodes dérivatives sont les plus utilisées dans la détection des transitions d'intensités. Elles consistent à calculer la dérivée en chaque point de l'image afin de mettre en évidence les variations de niveau de gris. On peut classer les méthodes dérivatives en deux groupes selon qu'on utilise la dérivée première (approche gradient) ou la dérivée seconde (approche Laplacien). L'allure de la dérivée première et de la dérivée seconde d'un signal de type « saut d'amplitude », qui peut correspondre à une variation locale et dans une seule direction du niveau de gris, est donnée par la figure (1.1) suivante.

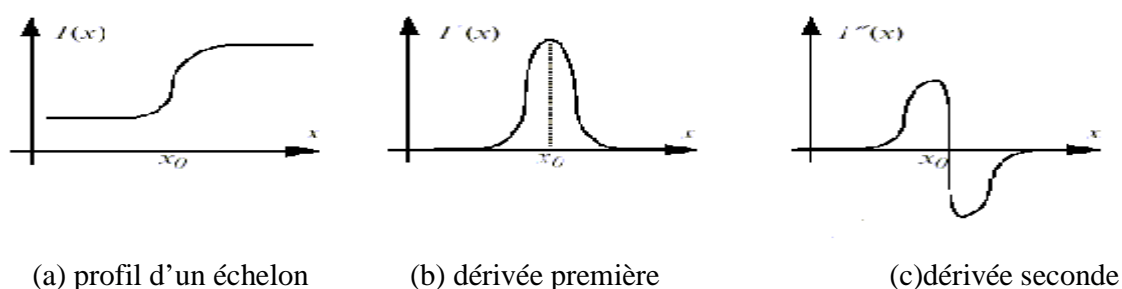


Fig.1.1: Dérivées première et seconde d'un contour de type "saut d'amplitude"

Une zone de transition dans le signal correspond à un maximum (ou un minimum) local de la dérivée première et un passage par zéro de la dérivée seconde. L'identification d'une zone de transition dans l'image peut être ainsi faite par seuillage de la norme de sa dérivée première ou du passage par zéro de sa dérivée seconde. Parmi les opérateurs gradient les plus connus, on trouve les masques de Robert [Roberts, 1965], de Prewitt [Prewitt, 1970], de Sobel [Sobel, 1978], de Kirsh [Kirsh, 1971].

Les méthodes dérivatives sont faciles à implémenter, rapides en termes de temps de calcul et donnent généralement de bons résultats sur des images non bruitées. Elles restent cependant sensibles aux petites variations non significatives des niveaux de gris (bruit) et produisent des contours discontinus qu'il faut ensuite fermer. Pour améliorer la qualité des méthodes dérivatives et pallier aux problèmes de précision et de localisation des contours, des opérateurs de dérivation avec filtrages optimaux ont été introduits.

Un Filtre optimal est un filtre dérivateur qui permet de détecter des contours en respectant les 3 critères suivants [Coquerez et al., 1995]:

- Une bonne détection : l'opérateur donne une réponse au voisinage d'un contour.
- Une bonne localisation : optimisation de la précision avec laquelle le contour est détecté.
- Unicité de la réponse : le contour doit provoquer une réponse unique de l'opérateur.

Plusieurs filtres optimaux sont apparus dans la littérature. Parmi eux, on trouve les filtres de Canny [Canny, 1986], de Deriche [Deriche, 1987] et [Deriche, 1990] et celui de Shen et Castan [Shen et al., 1986], [Castan et al., 1989], [Shen et al., 1992].

Quant aux contours actifs (Snakes en anglais), ils sont plus récents, ils ont été introduits par Kass et al. [Kass et al., 1988] et sont définies comme des courbes capable de se déformer et de se déplacer pour épouser la forme des objets constituant l'image [Azerradj et Kessoum, 2007]. L'idée de cette méthode est de déplacer les points de la courbe pour les rapprocher des zones de fort gradient tout en conservant des caractéristiques comme la courbure du contour ou la répartition des points sur le contour ou d'autres contraintes liées à la disposition des points [Chilali, 2006]. Les contours actifs permettent de détecter des contours fermés mais restent sensibles, à l'instar des méthodes dérivatives, aux variations locales des niveaux de gris. La figure (1.2) illustre l'évolution de contour actif avec $\Gamma(t)$ est le contour initial qui peut être défini manuellement et v la vitesse d'évolution de la courbe.

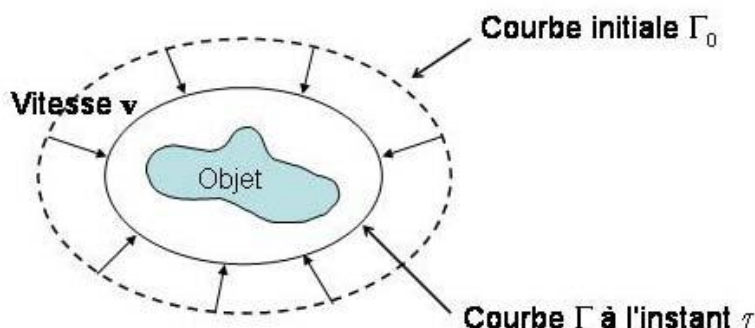


Fig.1.2: Evolution du contour actif $\Gamma(t)$ avec la vitesse v vers l'objet d'intérêt

1.3.2. Approche région

Contrairement à la segmentation d'images par l'approche contour, la segmentation par régions consiste à décomposer l'image en régions homogènes [Gonzalez et Woods., 1992]. Ces régions sont composées d'ensemble des pixels connexes possédant des propriétés au sens d'un prédicat d'homogénéité donné. On distingue les méthodes qui tiennent compte de la connexité des pixels et celles qui n'utilisent pas cette information.

Dans le premier cas, en plus de l'information niveau de gris, l'information spatiale est automatiquement intégrée dans le processus de segmentation [Pavlidis et Liow, 1990], le contrôle géométrique peut être aussi utilisé pour contrôler ce processus. Le problème fondamental associé à cette approche est lié à la définition du critère d'homogénéité utilisé dans le processus de segmentation de l'image. On distingue quatre types de méthodes.

1.3.2.1. Croissance des régions

Cette approche consiste à choisir initialement un ensemble de pixels pour les considérer comme des germes ou noyaux des régions à partir desquels la croissance des régions s'effectuera. Chaque région doit respecter un critère d'uniformité ou d'homogénéité défini par exemple par la variance des niveaux de gris. La croissance des régions s'effectue par agrégation de nouveaux pixels aux germes initiaux. Elle consiste à fusionner successivement aux régions les pixels voisins qui sont similaires et à condition que les régions fusionnées continuent à vérifier le critère d'uniformité. Notant que des régions adjacentes peuvent être fusionnées si leur fusion vérifie le critère d'uniformité et que de nouveaux germes peuvent être créés dans les régions qu'elles ne pouvant pas être fusionnées avec les régions existantes [Boucher, 1998].

1.3.2.2. Division des régions

Les modèles de division de régions consistent à diviser l'image d'une manière récursive tant qu'un critère d'homogénéité sur les régions n'est pas vérifié. Initialement, l'image est divisée en zones. Chaque zone ainsi créée est elle-même divisée si elle ne vérifie pas le critère d'homogénéité. Ce processus de division se poursuit pour toutes les zones nouvellement créées et s'arrête que lorsque toutes les zones sont homogènes ou bien leurs tailles sont en dessous d'un seuil fixé préalablement. La division de l'image est généralement représentée par une structure géométrique bien particulière telle que la structure à arbre quaternaire « quadtree » ou la structure de Voronoï. La figure (1.3) montre le principe de la segmentation par division suivant la structure d'arbre quaternaire.

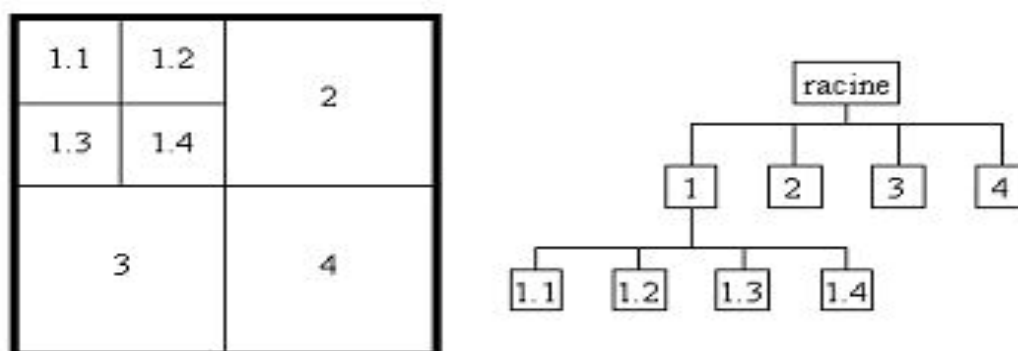


Fig. 1.3: Principe de la segmentation par division des régions (Structure quaternaire)

Le critère d'homogénéité peut être évalué par la variance des niveaux de gris ou peut être défini à partir de l'allure l'histogramme de niveau de gris. Par exemple dans [Ohlander, 1979], l'histogramme de chaque région est analysé, si il est unimodal (région homogène), la région reste inchangée, sinon elle est divisée. La division prend fin quand toutes les régions ont un histogramme unimodal.

1.3.2.3. Division/fusion

Ces méthodes combinent successivement une méthode de division de l'image en des petites régions homogènes comme celle décrite précédemment et une méthode de fusion des régions connexes et similaires au sens d'un prédicat de regroupement. Le processus de segmentation utilise la structure pyramidale telle que les arbres quaternaires. Au cours de la phase de fusion, on progresse de manière récursive vers le haut de l'arbre en fusionnant les régions qui forment au niveau supérieur une région homogène. Horowitz et Pavlidis [Horwitz

and Pavlidis, 1976], sont les premiers à avoir proposé une telle approche de segmentation à partir de laquelle de nouvelles variantes sont apparues dans la littérature [Beveridge et al., 1989], Strasters and Gerbrands, 1991], [Gonzalez, 1992]. La figure (1. 4) illustre le principe de la division/fusion. Les résultats dépendent de l'ordre dans lequel les régions sont examinées pour être éventuellement regroupées.

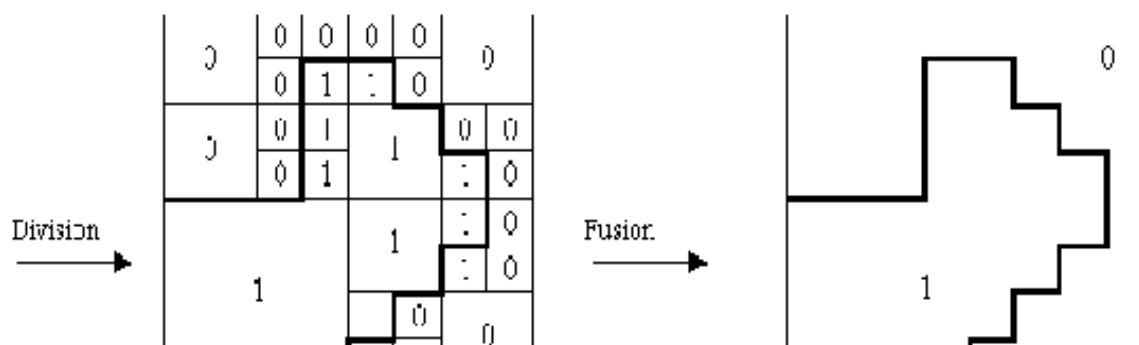


Fig. 1.4: Principe de la segmentation par division-fusion.

1.3.2.4. Segmentation d'image par classification

La classification est l'une des procédures les plus utilisées en analyse des données, elle permet de partitionner un ensemble de données multidimensionnelles en un ensemble de k classes disjointes. En segmentation d'images, les données multidimensionnelles correspondent aux pixels de l'image ou chaque pixel est caractérisé par un vecteur d'attributs tels que les attributs de texture ou les composantes couleurs. Chaque classe regroupe des pixels ayant des vecteurs de caractéristiques aussi similaires que possible. Sachant que les pixels de deux classes distinctes ont des attributs très différents [Jain et al., 1999]. Contrairement aux méthodes précédentes, les techniques de segmentation par classification ne prennent pas en compte la disposition spatiale des pixels et ne considèrent que le vecteur de caractéristiques. Elles identifient les classes de pixels présentes dans l'image et affectent à chaque pixel une étiquette indiquant la classe à laquelle il appartient. Ainsi les pixels appartenant à une classe peuvent former plusieurs régions non adjacentes dans l'image mais partageant les mêmes propriétés. Une segmentation en régions n'est obtenue qu'après analyse de la connexité des pixels dans l'image étiquetée.

On trouve dans la littérature une multitude de méthodes de classification parmi lesquelles on peut citer, l'algorithme K-means et sa version floue fuzzy-C-means, l'algorithme d'estimation-maximisation (EM), l'algorithme mean-shift, etc.

Lorsqu'on considère le niveau de gris comme la seule caractéristique pour représenter un pixel, la segmentation par classification peut être abordée par des techniques de seuillage qui seront présentées en détail dans les sections suivantes et dans le prochain chapitre.

1.3.3. Approche coopérative

L'approche contour et l'approche région sont en fait des approches duales, on peut alors les combiner afin d'aboutir à une segmentation plus efficace. Comme, il est possible de combiner plusieurs techniques d'une même approche. L'intérêt de faire coopérer plusieurs approches permet de définir automatiquement des ensembles de contraintes pour mieux guider le processus de segmentation. Dans une approche coopérative, la stratégie de résolution peut être séquentielle, c'est-à-dire les résultats d'une approche sont exploités successivement pour en guider une autre, ou bien itérative, De plus, des principes de fusion d'information peuvent être exploités pour renforcer des critères de décision et influencer les choix de segmentation [Laurence, 1999]. Dans le cadre de la classification, par exemple, la nécessité de définir une région d'intérêt est fondamentale pour réduire le nombre de classes à classifier. De même, le choix d'une stratégie de résolution, lorsqu'elle n'est pas guidée par un ensemble de connaissance à priori n'est pas évident.

Il existe de multiples façons d'envisager la coopération contour-région [Pavlidis et Liow, 1990]. Le terme de «coopération contour-région» signifie l'utilisation des informations des contours pour autoriser ou non la fusion de deux régions ou la division d'une région; réciproquement, il peut signifier l'utilisation d'informations sur les régions pour autoriser ou non la fusion de deux contours ou la croissance d'un contour. Il existe aussi une coopération région-région quand deux régions coopèrent pour fusionner ou une coopération contour-contour quand deux contours coopèrent pour se connecter. Dans tous les cas, il existe de multiples critères possibles (par exemple, écart des niveaux de gris entre deux régions, pourcentage de points de contours à la frontière de deux régions,...) pour autoriser la coopération ainsi que de multiples manières permettant de les combiner [Laurence, 1999].

Il existe également d'autres méthodes de segmentation qui n'appartiennent pas aux catégories citées précédemment et qui utilisent des formalismes différents. A titre d'exemple, on peut citer la méthode des contours actifs basée région [Nakib, 2007. 1], les réseaux de neurones comme les cartes organisatrices de Kohonen [Kohonen, 1989], [Thiria et al., 1997] connu sous le nom de SOM (Self Organising Map), la morphologie mathématique représenté par l'algorithme

de lignes de partage des eaux (Watershed) Digabel et al [Digabel et al., 1978] [Najman and Couprie, 2003] ...etc.

1.4. Segmentation par seuillage

La segmentation par seuillage d'histogramme constitue un cas particulier de la segmentation par classification. Elle permet de répartir les pixels en classes en fonction de leurs niveaux de gris. Les classes sont alors délimitées par des seuils.

1.4.1. Définition du seuillage

Soit $\{0,1,2,\dots,L-1\}$ l'ensemble des niveaux de gris d'une image, $l(x,y)$ la luminance (niveau de gris) d'un pixel de coordonnées (x,y) , $L-1$ étant le niveau de gris maximal. (Souvent $L = 256$). La segmentation par seuillage est une opération qui consiste à répartir les pixels en K classes (C_1, C_2, \dots, C_K) à partir d'un ensemble de seuils $T = \{t_1, t_2, \dots, t_{K-1}\}$. Par convenance, on utilise deux autres seuils, $t_0 = 0$ et $t_K = L-1$. Un pixel de niveau de gris $l(x,y)$ est affecté à la classe C_k si $t_k \leq l(x,y) \leq t_{k+1}$ avec $k = 0,1,2,\dots,K-1$.

Pour une image comportant des objets clairs sur un fond sombre, le seuillage se résume à choisir un niveau de gris t (un seul seuil) et à affecter à chaque pixel de l'image le label objet ou fond selon le test suivant :

§ Si $l(x,y) < t$ alors le pixel est étiqueté «fond», sinon le pixel est étiqueté «objet».

Dans ce cas, on parle du seuillage tout court (bilevel thresholding) et l'image segmentée ainsi obtenue est appelée image binaire.

§ Si par contre, le nombre de seuils recherchés est supérieur à 1, on parle de multiseuillage ou multi-level thresholding.

Plusieurs techniques ont été proposées pour le calcul d'un seul seuil puis étendues au problème du multiseuillage. Cependant, en pratique cette extension peut engendrer des temps de calculs prohibitifs. En effet la recherche rapide d'un seul seuil d'une manière exhaustive est tout à fait possible. Cependant cette recherche exhaustive devient prohibitive lorsque le nombre de seuils augmente. Chang et Al notent que la complexité des calculs augmente exponentiellement lorsque le nombre de seuils augmente [Chang et al., 1995]. Pour K classes le nombre d'opérations nécessaires pour le calcul de $K-1$ seuils est donné par :

$$NO = \frac{(L + K)!}{L!K!}$$

Pour résoudre ce problème plusieurs techniques ont été proposées. Parmi elles on peut citer l'algorithme itératif proposé par Yin et Chen [Yin and Chen, 1997], ou encore des techniques d'optimisation métaheuristiques [Horng, 2010].

Les méthodes de seuillage peuvent être réparties en deux catégories selon qu'on définit un même seuil pour tous les pixels de l'image (seuillage globale) ou un seuil pour chaque pixel (seuillage local ou adaptatif).

1.4.2. Méthodes de seuillage globale

Les méthodes de seuillage globale reposent sur l'exploitation de l'histogramme de toute l'image. L'histogramme est une courbe monodimensionnelle qui caractérise la distribution des niveaux de gris, il est décrit par une fonction monovariante discrète $h(i)$ ou $p(i)$ qui représente respectivement la fréquence ou la probabilité d'apparition du niveau de gris i , tel que :

$$p(i) = \frac{h(i)}{N}$$

$h(i)$ étant le nombre de pixels ayant le niveau de gris i et N le nombre total de pixels dans l'image.

A partir de l'allure de l'histogramme, on peut déduire le nombre de classes ainsi que la position relative des seuils (Fig. 1.5). L'histogramme est unimodal, s'il est formé d'un seul pic représentant les pixels de l'objet ou ceux du fond. Il est bimodal s'il est caractérisé par deux modes séparés par une vallée, il indique l'existence d'un objet sur un fond. L'histogramme multimodal comporte plus de deux modes séparés par des vallées, il indique la présence de plusieurs classes dans l'image (chaque mode correspond à une classe). Les seuils doivent être localisés dans les vallées.

En pratique, il est rare de trouver un histogramme qui présente des modes bien distincts. En effet, la plupart des images présentent des histogrammes bruités caractérisés par des modes non discernables.

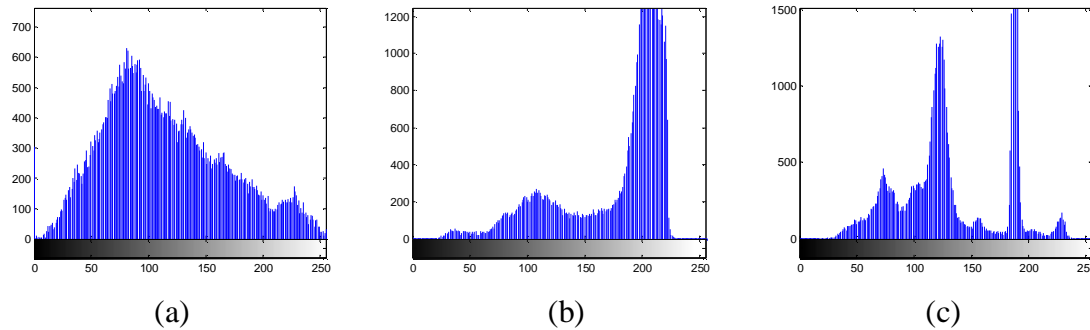


Fig. 1.5: Allures idéales d'un histogramme. (a): unimodal, (b): bimodal et (c) multimodal

Il existe toute une panoplie de méthodes de seuillages globales qu'on peut répartir en deux catégories : méthodes paramétriques et méthodes non paramétriques.

1.4.2.1. Méthodes paramétriques

Les méthodes de seuillage paramétrique supposent l'histogramme ($p(i)$) peut être approximé par une combinaison linéaire de fonctions de densité de probabilité (fdp) ($p_a(i)$) dont le modèle est connu à priori tel que :

$$p_a(i) = \sum_{k=1}^K P_k p(i, q_k)$$

Où $p(i, q_k)$ est la fonction de probabilité de la classe C_k ayant les paramètres statistiques q_k et P_k la probabilité à priori de la classe C_k et qui peut être estimé par le nombre de pixels de la classe C_k sur le nombre total N de pixels. $P_k = \frac{h(i)}{N} \Big|_{i \in C_k}$.

Le problème du seuillage revient donc à estimer les probabilités à priori P_k et les paramètres statistiques q_k de chaque fdp ou les seuils t_k en minimisant l'erreur quadratique moyenne suivante:

$$J(T) \equiv J(\Theta) = \frac{1}{L} \sum_{i=0}^{L-1} [p(i) - p_a(i)]^2$$

Souvent, les fonctions de densités de probabilité sont supposées suivre un modèle gaussien de moyenne m_k et de variance S_k tel que :

$$p(i, q_k) = \frac{1}{\sqrt{2\pi} s_k} e^{-\frac{(i-m_k)^2}{s_k^2}}$$

P_k , m_k et s_k peuvent être estimés par :

$$P_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} h(i), \quad m_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} ih(i), \quad s_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} (i - m_k)^2 h(i)$$

Le vecteur paramètre recherché est défini par : $\Theta = \{P_k, m_k, s_k\}$ avec $k = 1, 2, \dots, K$.

La minimisation de la fonctionnelle $J(\Theta)$ ou $J(T)$ peut être résolue par l'algorithme d'estimation-maximisation [Dempster et al., 1977] ou en utilisant les méthodes d'optimisation métaheuristique telles que celles présentées dans le chapitre 3.

1.4.2.2. Méthodes non paramétriques

Les méthodes non paramétriques de seuillage permettent de trouver les seuils optimaux sans tenir compte d'aucune hypothèse sur la forme de l'histogramme. Ces méthodes sont très nombreuses, elles constituent la grande majorité des méthodes de seuillage et sont généralement basées sur l'optimisation d'une fonction objective. Nous présenterons dans ce qui suit quelques unes d'entre elles.

1.4.2.2.1. Méthode d'Otsu

Elle est considérée comme la méthode de référence dans le domaine du seuillage d'histogrammes. Dans cette méthode [Otsu, 1979], l'opération de seuillage est vue comme une séparation (un partitionnement) des pixels d'une image en deux classes C_1 (fond), C_2 (objet) à partir d'un seuil t . La classe « fond » regroupe tous les pixels ayant un niveau de gris inférieur au seuil t alors que la classe « objet » contient tous les pixels de niveau de gris supérieur à t . Ces deux classes peuvent être désignées en fonction du seuil t comme suit :

$$C_1 = \{0, 1, \dots, t\} \text{ et } C_2 = \{t+1, \dots, L-1\}$$

Soient s_w^2 la variance d'une classe, s_b^2 la variance interclasse et s_T^2 la variance totale telles que :

$$s_B^2 = P_1 P_2 (m_2 - m_1)^2, \quad s_T^2 = \sum_{i=1}^{L-1} p_i (i - m)^2, \quad s_w^2(t) = \sum_{i=0}^t P_1 (i - m_1)^2 + \sum_{i=t+1}^{L-1} P_2 (i - m_2)^2$$

$$S_T^2 = S_B^2 + S_w^2$$

m_1 , m_2 et m désignent respectivement les niveau de gris moyen des classes C_1 , C_2 et de

$$\text{l'image tels que : } m_1 = \sum_{i=1}^t i \cdot \frac{p_i}{P_1}, \quad m_2 = \sum_{i=t+1}^{L-1} i \cdot \frac{p_i}{P_2}, \quad m = \sum_{i=1}^{L-1} i \cdot p_i,$$

P_1 et P_2 représentent respectivement les probabilités à priori des classes C_1 et C_2 tels que :

$$P_1 = \sum_{i=1}^t p_i, \quad P_2 = \sum_{i=t+1}^{L-1} p_i \quad \text{et} \quad P_1 + P_2 = 1$$

Le seuil optimum t^* peut être déterminé en maximisant un des trois critères suivant :

$$l(t) = \frac{S_B^2}{S_w^2}; \quad h(t) = \frac{S_B^2}{S_T^2}; \quad k(t) = \frac{S_T^2}{S_w^2}$$

Les trois critères sont équivalents, mais pour des raisons de simplicité, la variance interclasse est la plus utilisée. Le seuil optimal t^* est celui qui maximise cette variance :

$$t^* = \underset{0 \leq t \leq L-1}{\text{Arg max}} S_B^2$$

Dans le cas du multiseuillage, la méthode d'Otsu peut être étendue au calcul d'un ensemble de seuils $T = (t_1, t_2, \dots, t_k, \dots, t_{K-1})$ en maximisant la variance interclasses.

$$T^* = \underset{0 \leq t_k \leq L-1}{\text{Arg max}} J(T)$$

$$\text{Où } J(T) = S_B^2 = \sum_{k=1}^K P_k (m_k - m)^2$$

avec P_k et m_k les probabilités et la moyennes respectives de la classes C_k tels que :

$$P_k = \sum_{i=t_{k-1}}^{t_k-1} p_i \quad \text{et} \quad m_k = \sum_{i=t_{k-1}}^{t_k-1} i p_i$$

1.4.2.2. Méthodes de Kittler et Illingworth

Proposée par Kittler et Illingworth [Kittler et al., 1986], cette méthode considère l'histogramme des niveaux de gris comme une estimée de la fonction de densité de probabilité $p(i)$ d'un mélange de populations formées des niveaux de gris des objets et du fond. On peut

supposer que chacune des deux composantes du mélange forment une distribution normale avec une moyenne m_i , un écart type s_i et une probabilité à priori p_i . On a ainsi :

$$p\left(\frac{i}{C_k}\right) = \frac{1}{\sqrt{2\pi} s_k} \exp\left(-\frac{(i - m_k)^2}{2s_k^2}\right)$$

$$p(i) = \sum_{k=0}^1 P_k \cdot P\left(\frac{i}{C_k}\right)$$

Le seuil optimal est obtenu en résolvant l'équation quadratique ci-dessous :

$$\frac{(i - m_0)^2}{s_0^2} + \text{Log } s_0^2 - 2 \text{Log } P_0 = \frac{(i - m_1)^2}{s_1^2} + \text{Log } s_1^2 - 2 \text{Log } P_1$$

ou en minimisant soit la fonction $J_1(t)$ suivante :

$$J_1(t) = I + 2 [P_0(t) \text{Log } s_0(t) + P_1 \text{Log } s_1(t)] - 2 [P_0(t) \text{Log } P_0(t) + P_1(t) \text{Log } P_1(t)]$$

avec:

$$s_0^2(t) = \frac{\sum_{i=0}^t (i - m_0(t))^2 h(i)}{P_0(t)}, \quad s_1^2(t) = \frac{\sum_{i=t+1}^{l-1} (i - m_1(t))^2 h(i)}{P_1(t)}$$

$$P_0(t) = \sum_{i=0}^t h(i) \quad \text{et} \quad P_1(t) = \sum_{i=t+1}^{l-1} h(i);$$

$$m_0(t) = \frac{\sum_{i=0}^t i h(i)}{P_0(t)} \quad \text{et} \quad m_1(t) = \frac{\sum_{i=t+1}^{l-1} i h(i)}{P_1(t)}.$$

Soit la fonction de critère $J_2(t)$.

$$J_2(t) = N^2 \left(w_0 \log \frac{s_0}{w_0 N^2} + w_1 \text{Log} \frac{s_1}{w_1 N^2} \right)$$

avec

$$s_0^2 = \frac{1}{w_0 N^2} \sum_{i=0}^t (i - m_0)^2 h_i(i), \quad s_1^2 = \frac{1}{w_1 N^2} \sum_{i=t}^{l-1} (i - m_1)^2 h(i)$$

Les paramètres w_0, w_1, m_0, m_1 sont calculés comme suit :

$$w_0 = \sum_{0 \leq i \leq t} p_i, \quad m_0 = \sum_{0 \leq i \leq t} i \cdot \frac{p_i}{w_0} \quad \text{et} \quad w_1 = \sum_{t \leq i \leq l-1} p_i, \quad m_1 = \sum_{t \leq i \leq l-1} i \cdot \frac{p_i}{w_0}$$

1.4.2.2.3. Méthode de Kapur

La méthode de Kapur et al [Kapur et al., 1985] est basée sur le principe de maximisation de l'entropie de Shannon. Elle suppose que les classes « objet » et « fond » possèdent deux densités de probabilité indépendantes.

$$C_1 : \frac{p_0}{P_1}, \frac{p_1}{P_1}, \dots, \frac{p_t}{P_1}$$

$$C_2 : \frac{p_{t+1}}{P_2}, \frac{p_{t+2}}{P_2}, \dots, \frac{p_{L-1}}{P_2}$$

où $p_i = \frac{h(i)}{N}$, $P_1 = \sum_{i=0}^t p_i$ et $P_2 = \sum_{i=t+1}^{L-1} p_i$ avec $P_1 + P_2 = 1$

Les entropies de Shannon associées aux deux distributions sont :

$$H_1 = -\sum_{i=0}^t \frac{p_i}{P_1} \log\left(\frac{p_i}{P_1}\right) \text{ et } H_2 = -\sum_{i=t+1}^{L-1} \frac{p_i}{P_2} \log\left(\frac{p_i}{P_2}\right)$$

Le seuil optimal t^* est alors défini comme étant le niveau de gris qui maximise les deux entropies, c'est-à-dire que :

$$t^* = \arg \max_{0 \leq t \leq L-1} \{H_1(t) + H_2(t)\}$$

Cette méthode peut être également étendue au calcul de plusieurs seuils $T = \{t_1, t_2, \dots, t_{k-1}\}$, Il s'agira alors de maximiser le critère suivant :

$$J(T) = \sum_{k=1}^K H_k$$

avec $H_k = \sum_{i=t_{k-1}}^{t_k-1} \frac{p_i}{P_k} \log\left(\frac{p_i}{P_k}\right)$ et $P_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} h(i)$

1.4.2.2.4. Méthode de cross entropie

Cette méthode a été introduite par Li et al. [Li et al., 1993], elle consiste à minimiser l'entropie entre l'image et sa version binaire, également appelée « cross entropie ». Le seuil optimal t^* peut être obtenu en maximisant les cross entropies des deux classes « objet » et « fond »

$$t^* = \arg \max_{0 \leq t \leq L-1} \{H_1(t) + H_2(t)\}$$

Ou $H_1(t) = \sum_{i=0}^t ih(i) \log \frac{i}{\mu_1}$ et $H_2(t) = \sum_{i=t+1}^{l-1} ih(i) \log \frac{i}{\mu_2}$ correspondent aux cross entropies des deux classes.

m_1 et m_2 correspondent aux moyennes de chaque classe tels que :

$$m_1 = \sum_{i=0}^t ih(i) \text{ et } m_2 = \sum_{i=t+1}^{l-1} ih(i)$$

Après quelques modifications, l'expression de $J(t)$ deviendra :

$$H_1(t) + H_2(t) = N^2 (P_1 m_1 \log(m_1) + P_2 m_2 \log(m_2))$$

$$\text{avec } P_1 = \sum_{i=0}^t p_i, \quad P_2 = \sum_{i=t+1}^{l-1} p_i, \quad m_1 = \sum_{i=0}^t i \frac{p_i}{w_0} \text{ et } m_2 = \sum_{i=t+1}^{l-1} i \frac{p_i}{w_1}$$

Dans le cas du multiseuillage, le critère à maximiser est:

$$J(T) = N^2 \sum_{k=1}^K P_k m_k \log(m_k)$$

$$\text{avec } P_k = \sum_{i=t_{k-1}}^{t_k-1} p_i \text{ et } m_k = \sum_{i=t_{k-1}}^{t_k-1} i \frac{p_i}{w_k} = \frac{m_k}{w_k}$$

1.4.2.2.5. Méthode de corrélation entropique

Proposée par Chang et al [Chang et al., 1995], cette méthode maximise la corrélation entropique entre les classes « objet » et « fond ». Soit X une variable aléatoire discrète, avec un rang $b = x_0, x_1, \dots$, fini ou infini et P_i la probabilité de $X = x_i$. La corrélation de X est définie comme suit :

$$R_X(t) = -\ln \sum_{i \geq 0} P_i^2$$

En se basant sur cette définition, les corrélations associées aux classes C_1 (fond) et C_2 (objet) sont données par les relations suivantes :

$$R_1(t) = -\ln \sum_{i=0}^t \left(\frac{P_i}{P_1} \right)^2 \text{ et } R_2(t) = -\ln \sum_{i=t+1}^{l-1} \left(\frac{P_i}{P_2} \right)^2$$

La corrélation totale associée aux deux classes est alors :

$$J(t) = R_1(t) + R_2(t) = -\ln \sum_{i=0}^t \left(\frac{P_i}{P_1} \right)^2 - \ln \sum_{i=t+1}^{l-1} \left(\frac{P_i}{P_2} \right)^2$$

$$= -\ln\left(\frac{G_1(t)G_2(t)}{P_1^2 P_2^2}\right) = -\ln(G_1(t)G_2(t)) + 2\ln(P_1 P_2)$$

avec

$$G_1(t) = \sum_{i=0}^t p_i^2, \quad G_2(t) = \sum_{i=t+1}^{l-1} p_i^2, \quad P_1 = \sum_{i=0}^t p_i \quad \text{et} \quad P_2 = \sum_{i=t+1}^{l-1} p_i$$

Le seuil optimal t^* est déterminé en maximisant la fonction $J(t)$

$$t^* = \arg \max_{0 \leq t \leq L-1} \{J(t)\}$$

Une extension de cette méthode au cas du multi seuillage consiste à déterminer l'ensemble des seuils $T = \{t_1, t_2, \dots, t_{K-1}\}$ qui maximise la fonction critère suivante

$$J(T) = -\ln \prod_{k=1}^K (G_k) + 2\ln \prod_{k=1}^K P_k$$

avec $G_k(t) = \sum_{i=t_{k-1}}^{t_k-1} p_i^2$ et $P_k = \sum_{i=t_{k-1}}^{t_k-1} p_i$

1.4.2.2.6. Méthode d'entropie de Renyi

Proposée par Sahoo et al, [Sahoo et al., 1997], cette méthode est une extension de la méthode de Kapur et celle de Chung [Chung et al., 2005]. Elle utilise deux distributions de probabilités (objet et fond) tirés de la distribution originale

$$C_0 : \frac{p_0}{P(C_0)}, \frac{p_1}{P(C_0)}, \dots, \frac{p_t}{P(C_0)}, \quad C_1 : \frac{p_{t+1}}{P(C_1)}, \frac{p_{t+2}}{P(C_1)}, \dots, \frac{p_{l-1}}{P(C_1)}$$

avec

$$P(C_0) = \sum_{i=0}^t p_i, \quad P(C_1) = \sum_{i=t+1}^{l-1} p_i \quad \text{et} \quad P(C_0) + P(C_1) = 1$$

L'entropie a priori de Renyi d'ordre a est définie par :

$$H_T^a = \frac{1}{1-a} \ln \sum_{i=0}^{l-1} (p_i)^a$$

$a (\neq 1)$ un paramètre réel positif.

L'entropie de Renyi constitue une généralisation de l'entropie de Shannon, puisque $\lim_{a \rightarrow 1} H_T^a = H_T$. Les entropies de Renyi associés aux classes fond et objet sont

$$H_{C_0}^a(t) = \frac{1}{1-a} \ln \sum_{i=0}^t \left(\frac{p_i}{P(C_0)} \right)^a \quad \text{et} \quad H_{C_1}^a(t) = \frac{1}{1-a} \ln \sum_{i=t+1}^{L-1} \left(\frac{p_i}{P(C_1)} \right)^a$$

soit t^* la valeur qui maximise $H_{C_0}^a(t) + H_{C_1}^a(t)$ tel que :

$$t^* = \underset{t \in L}{\text{Arg max}} \{H_{C_0}^a(t) + H_{C_1}^a(t)\}$$

Une extension de cette méthode au cas du multi seuillage consiste à déterminer l'ensemble des seuils $T = \{t_1, t_2, \dots, t_{k-1}\}$ qui maximise la fonction critère suivante :

$$J(T) = \sum_{i=t_{k-1}}^{t_k-1} H_T^a$$

avec $H_T^a = \frac{1}{1-a} \sum_{i=t_{k-1}}^{t_k-1} \left(\frac{p_i}{P_k} \right)^a$ et $P_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} h(i)$

$$\{t_1^*, t_2^*, \dots, t_{k-1}^*\} = \arg \max \{J(T)\}$$

1.4.2.2.7. Méthode d'entropie de Tsallis-Havrda-Charvat

L'entropie de Tsallis [Sahoo et al., 2006] d'un système physique non extensif est définie par :

$$S^q = \frac{1 - \sum_i^k p_i^q}{q-1}$$

où q est un paramètre réel appelé indice entropique et k le nombre de possibilités du système. L'application de cette mesure d'entropie pour le seuillage d'histogramme consiste à maximiser l'entropie de Tsallis des classes « fond » et « objet » définies comme suit:

$$S_1^q(t) = \frac{1 - \sum_{i=1}^t \left(\frac{p_i}{P_1} \right)^q}{q-1} \quad \text{et} \quad S_2^q(t) = \frac{1 - \sum_{i=t+1}^{L-1} \left(\frac{p_i}{P_2} \right)^q}{q-1}$$

L'entropie totale de Tsallis s'exprime comme suit :

$$S^q(t) = [S_1^q + S_2^q + (1-q)S_1^q(t)S_2^q(t)]$$

Sa maximisation permet de déterminer le seuil optimal t^* .

$$t^* = \arg \max_{0 \leq t \leq L-1} \{S^q(t)\}$$

La qualité de la segmentation est conditionnée par la valeur de q [Nakib et al., 2007. 1], [Portes et al., 2004].

Une extension de cette méthode au cas du multi seuillage consiste à déterminer l'ensemble des seuils $T = \{t_1, t_2, \mathbf{K}, t_{k-1}\}$ qui maximise la fonction critère suivante :

$$S^q = \frac{1 - \sum_{i=t_{k-1}}^{t_k-1} p_i^q}{q-1}$$

$$T^* = \{t_1^*, t_2^*, \mathbf{K}, t_{k-1}^*\} = \arg \max \{S^q(t_1, t_2, \mathbf{K}, t_{k-1})\}$$

avec $S^q(t_1, t_2, \mathbf{K}, t_{k-1}) = [S_1^q + S_2^q + \mathbf{K} + S_{k-1}^q + (1-q)S_1^q S_2^q \mathbf{K} S_{k-1}^q]$

1.4.2.2.8. Méthode d'entropie exponentielle

L'entropie exponentielle est définie par : [Nakib et al., 2007. 3]

$$H^a = \left(\sum_i (p_i)^a \right)^{\frac{1}{1-a}}$$

Où $a \in \mathfrak{R}$ et $a \neq 1$ est un paramètre

Les entropies exponentielles de deux classes (fond et objet) sont données respectivement par les deux relations suivantes:

L'entropie exponentielle 1D de la classe 1 est donnée par :

$$H_1^a = \left(\sum_{i=s}^L \left(\frac{p_i}{P_1} \right)^a \right)^{\frac{1}{1-a}}$$

L'entropie exponentielle 1D de la classe 2 :

$$H_2^a(t) = \left(\sum_{i=t+1}^L \left(\frac{p_i}{P_2} \right)^a \right)^{\frac{1}{1-a}}$$

L'entropie totale est :

$$H^a(t) = H_1^a(t) + H_2^a(t)$$

Le seuil optimal t^* peut être déterminé en maximisant l'entropie exponentielle relative à la classe fond et à celle de l'objet:

$$t^* = \arg \max (H^a(t))$$

Une extension de cette méthode au cas du multi seuillage consiste à déterminer l'ensemble des seuils $T = \{t_1, t_2, \dots, t_{k-1}\}$ qui maximise la fonction critère suivante :

$$J(T) = \sum_{i=t_{k-1}}^{t_k-1} H_T^a$$

$$\text{avec } H_T^a = \left(\sum_{i=t_{k-1}}^{t_k-1} \left(\frac{P_i}{P_k} \right)^a \right)^{\frac{1}{(1-a)}} \text{ et } P_k = \frac{1}{N} \sum_{i=t_{k-1}}^{t_k-1} h(i)$$

1.4.3. Méthodes de seuillage locale ou adaptatif

Il existe des images pour lesquelles il est difficile d'effectuer une bonne binarisation au moyen d'un seul seuil, si celui-ci est le même pour chaque point de l'image. Ceci peut être dû à une non uniformité de l'éclairage de l'image ou au fait que les différents objets qui composent l'image ont des dynamiques de luminances différentes. Pour cela, les méthodes de binarisation locales sont mieux adaptées.

A l'inverse des méthodes globales qui ne considèrent que la valeur du pixel, les méthodes locales prennent en considération la valeur des pixels voisins pour le calcul des seuils. Deux stratégies sont souvent employées. La première consiste à diviser, en premier lieu, l'image en fenêtres non chevauchantes [Chehdi, 1991. 1] et [Chehdi, 1991. 2]. Dans un deuxième temps, un seuil est déterminé dans chaque fenêtre selon la nature de l'histogramme local. Un seuil égal à zéro est affecté à la fenêtre possédant un histogramme unimodal tandis qu'une procédure de seuillage est appliquée pour déterminer un seul seuil dans le cas d'un histogramme bimodal ou multimodal. Les seuils nuls sont finalement remplacés par les autres seuils non nuls par interpolation ou par moyennage. La figure (1.6) illustre le principe du seuillage local ou adaptatif [Nakib, 2007. 1].

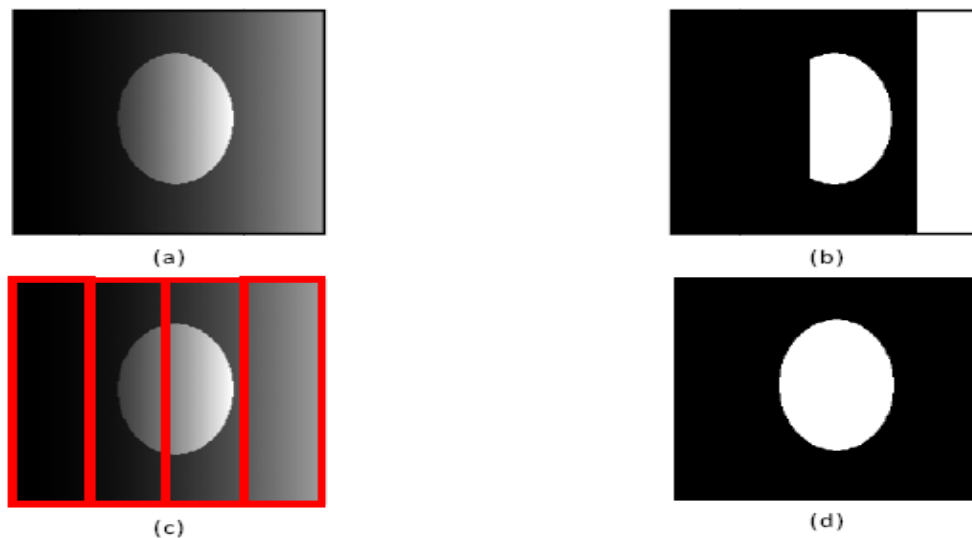


Fig. 1.6 : Illustration du seuillage adaptatif. (a) Image originale avec un fond non uniforme , (b) Image segmentée avec un seuil global, (c) Image originale découpée en sous images, (d) Image segmentée avec un seuillage adaptatif.

En ce qui concerne la deuxième stratégie, il s'agit de déterminer un seuil pour chaque pixel en fonction des niveaux de gris de ses pixels voisins situés dans une fenêtre de voisinage centrée sur le pixel courant. Parmi les méthodes utilisant cette stratégie, on peut citer les méthodes de Venkateswarlu [Venkateswarlu, 1995], de Bernsen [Bernsen, 1986], de Mardia et Hainsworth [Mardia and Hainsworth, 1988] et celle de Niblack [Niblack, 1986].

Dans cette dernière méthode, le calcul du seuil $t(x, y)$ de chaque pixel de l'image en niveau de gris $l(x, y)$ centré dans une fenêtre de voisinage de taille $(W \times W)$, est basé sur l'évaluation de la moyenne locale m et l'écart type s à partir des niveaux de gris des pixels de cette fenêtre tel que :

$$t(x, y) = m(x, y) + \alpha s(x, y)$$

avec $\alpha \in \mathfrak{R}$, un paramètre fixe.

$$m(x, y) = \frac{1}{W^2} \sum_{\substack{x-W/2 \leq x' \leq x+W/2 \\ y-W/2 \leq y' \leq y+W/2}} l(x', y') \quad \text{et} \quad s^2(x, y) = \frac{1}{(W^2 - 1)} \sum_{\substack{x-W/2 \leq x' \leq x+W/2 \\ y-W/2 \leq y' \leq y+W/2}} (l(x', y') - \mu(x, y))^2$$

Les méthodes de seuillage locale sont confrontées au choix de la taille de la fenêtre de voisinage. Celle ci doit être suffisamment petite pour préserver les détails locaux et suffisamment larges pour supprimer le bruit.

1.5. Conclusion

Nous avons présenté dans ce chapitre quelques notions sur la segmentation d'images. Il ressort que le domaine de la segmentation est tellement riche et diversifié qu'il est difficile de donner une étude exhaustive. Celles ci se déclinent en trois grandes approches. Nous nous sommes particulièrement intéressés à l'approche région dans laquelle les méthodes basées sur la classification et plus précisément de seuillage ont attiré notre attention. Les méthodes de seuillage d'histogrammes sont nombreuses, nous avons détaillé que quelques une d'entres elles, celles qui seront étudiées dans le prochain chapitre et qui concerne le seuillage d'histogrammes bidimensionnels et tridimensionnels.

Chapitre 2

Seuillage d'histogrammes bidimensionnel et tridimensionnel

2.1. Introduction

Les performances des méthodes basées sur le seuillage d'histogrammes monodimensionnel se dégradent rapidement lorsque les images à seuiller sont très bruitées. Afin d'éviter ce problème on assiste de plus en plus à l'émergence de nouvelles méthodes plus robustes. Celles-ci sont basées sur des histogrammes bidimensionnel ou tridimensionnel. L'intérêt d'utiliser un histogramme multidimensionnel au lieu d'un histogramme monodimensionnel a été mentionné dans plusieurs travaux [Nakib et al, 2007. 1.2.3.4], [Sahoo et al. 2004], [Sahoo et al, 2006] ...etc. Nous décrirons dans ce chapitre une revue de méthodes de seuillage basées sur des histogrammes bidimensionnels et tridimensionnels.

2.2. Histogrammes bidimensionnels

Un histogramme bidimensionnel est défini d'une manière générale comme une fonction à deux variables discrètes x_1 et x_2 indiquant la fréquence ou la probabilité d'apparition de deux valeurs de ses variables x_1 et x_2 . La variable x_1 correspond généralement au niveau de gris des pixels de coordonnées (x, y) noté $i = l(x, y)$ et la variable x_2 au niveau de gris noté $j = g(x, y)$ d'un pixel voisin, ou au niveau de gris moyen, minimal ou maximal de chaque pixel ou encore au gradient du pixel considéré.

On note par h_{ij} la fréquence d'apparition du couple de niveaux de gris (i, j) . L'histogramme bidimensionnel d'une image peut être alors représenté par la fonction de probabilités jointe p_{ij} donnée par :

$$p_{ij} = \frac{h_{ij}}{N^2} \quad \text{avec } i, j = 0, 1, 2, \dots, L-1 \quad \text{tel que } i = l(x, y) \text{ et } j = g(x, y) \text{ et } \sum_{i=1}^L \sum_{j=1}^L p_{ij} = 1$$

Il existe plusieurs manières de définir un histogramme bidimensionnel dans le cadre du seuillage d'histogrammes selon le choix des variables x_1 et x_2 .

2.2.1. Matrice de cooccurrence.

Dans ce cas, les variables x_1 et x_2 correspondent aux niveaux de gris de deux pixels voisins. Initialement introduite par Haralick [Haralick, 1973] pour l'analyse de la texture, la matrice de cooccurrence notée $M_{(d,f)}$ est une matrice dont les éléments constituent les fréquences relatives d'occurrence, de deux pixels voisins, de niveaux de gris respectifs i et j , séparés par une distance d avec une orientation f . Le nombre de matrices peut être assez large, selon le choix f et d . Les matrices de cooccurrence définies par Haralick [Haralick,

1973] correspondent aux orientations f égale à $0^\circ, \frac{p}{2}, p, \frac{3p}{2}$. Souvent, on définit des matrices de transition notée (T), à partir des matrices de cooccurrence pour la sélection du seuil. Les matrices de transition peuvent être définies comme suit:

$$T_h = M_{(1,0)}, \quad T_v = M_{\left(1, \frac{3p}{2}\right)}, \quad T_{vh} = T_v + T_h$$

Ahuja et Rosenfeld [Rosenfeld and Kak, 1981] définissent également la matrice de cooccurrence dans les quatre directions telle que :

$$M = M_{(1,0)} + M_{\left(1, \frac{p}{2}\right)} + M_{(1,p)} + M_{\left(1, \frac{3p}{2}\right)}$$

Due à l'homogénéité des régions, les pixels des objets ou ceux des fonds doivent contribuer, principalement, aux éléments proches de la diagonale de M . Egalement, les pixels proches d'un contour doivent contribuer aux éléments éloignés de la diagonale. Ainsi on peut définir deux histogrammes à partir de la matrice M .

- § Un histogramme basé sur les éléments proches de la diagonale de M , cet histogramme doit avoir une vallée profonde entre les niveaux de gris des objets et ceux du fond.
- § Un histogramme basé sur les éléments hors diagonale de M doit avoir un pic aigu entre les niveaux de gris des objets et ceux du fond.

Le seuil de l'image peut être pris comme le niveau de gris qui correspond le mieux à la vallée du premier histogramme et donc au pic du second histogramme, si on les superposait.

2.2.2. Histogramme bidimensionnel proprement dit

Contrairement à l'histogramme monodimensionnel qui représente juste la distribution de niveau de gris en fonction de la fréquence de leur apparition dans l'image, l'histogramme bidimensionnel tient compte à la fois du niveau de gris d'un pixel (variable x_1) et du niveau de gris moyen (variable x_2) de ce pixel définie comme une moyenne locale $g(x, y)$ telle que :

$$g(x, y) = \frac{1}{n^2} \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} l(x+i, y+j)$$

(nxn) étant la dimension de la fenêtre de voisinage généralement égale à (3X3). Ainsi chaque pixel (x, y) d'une image peut être représenté par la paire [$l(x, y), g(x, y)$].

2.2.3. D'autres histogrammes bidimensionnels

Dans [Z. Hong Qing, 2004], l'histogramme bidimensionnel est définie par le niveau de gris des pixels (variable x_1) et le niveau de gris gradient (variable x_2) de ces pixels sachant que le niveau de gris gradient est déterminé en appliquant l'opérateur de Sobel à l'image. Ma et al. [Ma et al., 2008] définissent l'histogramme bidimensionnel en tenant compte de l'image filtrée et de l'image gradient. Il est également possible de définir un histogramme bidimensionnel à partir du niveau de gris minimal (variable x_1) et du niveau de gris maximal (variable x_2) définie localement pour chaque pixel à partir de ses pixels voisins situés dans une fenêtre de voisinage de taille $(n \times n)$ [Zhang and Zhang, 2006] telle que:

$$k(x, y) = \min_{-n/2 \leq i, j \leq n/2} (I(x+i, y+j)) \text{ et } g(x, y) = \max_{-n/2 \leq i, j \leq n/2} (I(x+i, y+j))$$

La dimension $(n \times n)$ de la fenêtre de voisinage est généralement égale à (3×3) .

Dans ce cas, h_{ij} représente la fréquence d'apparition du couple de niveaux de gris (i, j) par l'ensemble des pixels de coordonnées (x, y) telle que $i = k(x, y)$ et $j = g(x, y)$.

2.3. Méthodes de seuillage basées sur les matrices de cooccurrence

Le seuillage basée sur les matrices de cooccurrence consiste à chercher le seuil t qui divise l'ensemble des pixels en deux classes C_1 et C_2 . En fonction du seuil, la matrice de cooccurrence M ou de transition T peut être découpée en quatre régions notées a, b, c et d comme le montre la figure (2.1).

- la zone a est formée des pixels appartenant à la classe C_1 .
- la zone b est formée des pixels appartenant à la classe C_2 .
- les zones c et d correspondent aux pixels de transition entre les deux classes C_1 et C_2 .

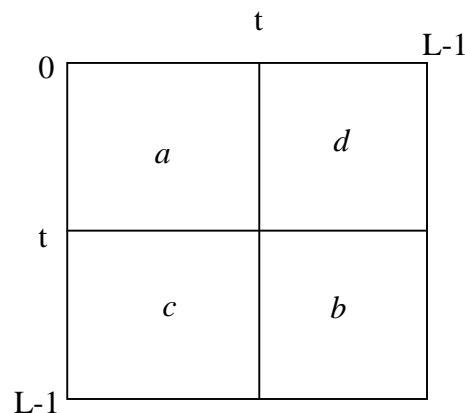


Fig. 2.1 : Seuillage d'une matrice cooccurrence

L'image binaire peut être alors obtenue comme dans le cas de l'histogramme monodimensionnel selon la règle suivante :

Si $l(x, y) < t$ alors le pixel est à la classe 1, sinon il est affecté à la classe 2.

Le choix du seuil t peut être fait de plusieurs manières. Weska et Rosenfeld [Weska, 1974], utilisent une matrice de transition symétrique définie par :

$$T = M_{(1,0)} + M_{\left(1, \frac{p}{2}\right)} + M_{(1,p)} + M_{\left(1, \frac{3p}{2}\right)}$$

Cette matrice T correspond à la somme des quatre matrices de cooccurrence au sens de Haralick dans les quatre directions, $0^\circ, 45^\circ, 90^\circ, 135^\circ$, pour une distance d de 1 pixel.

Le seuil optimal est celui qui maximise la quantité suivante :

$$P_j(t) = \frac{c + d}{a + b + c + d}$$

$P_j(t)$ correspond à la somme des valeurs rencontrées dans les zones c et d de la matrice de transaction. Plus $P_j(t)$ est grand, plus on trouve des pixels de transition entre les deux régions.

Deravi et Pal [Deravi et al., 1983], utilisent, quant à eux, une matrice non symétrique définie par :

$$T = M_{(1,0)} + M_{\left(1, \frac{p}{2}\right)}$$

Pour un niveau de gris t permettant de définir deux classes C_1 et C_2 , ils définissent les probabilités conditionnelles de transition $P_1(t)$ et $P_2(t)$ entre les deux classes.

$$P_1(t) = \frac{c}{a + c} \quad \text{et} \quad P_2(t) = \frac{d}{c + d}$$

$P_2(t)$ représente la probabilité de passer de la classe C_1 à la classe C_2 et $P_1(t)$ représente la probabilité de passer de la classe C_2 à la classe C_1 .

La probabilité de transition entre les classes séparées est alors:

$$P_c(t) = \frac{1}{2} \left\{ \frac{c}{a + c} + \frac{d}{b + d} \right\}$$

Deravi et Pal [Deravi et al., 1983] appellent $P_j(t)$ et $P_c(t)$ les mesures d'interaction, le seuil optimal est obtenu en maximisant $P_j(t)$ ou $P_c(t)$.

Chanda et Al utilisent une mesure de contraste moyen à partir d'une matrice T non symétrique définie comme suit : $T = M_{\left(1, \frac{p}{2}\right)}$

La mesure de contraste moyen est :

$$J(t) = \frac{\sum_{i=0}^t \sum_{j=t+1}^{L-1} T_{ij}(i-j)}{\sum_{i=0}^t \sum_{j=t+1}^{L-1} T_{ij}} + \frac{\sum_{i=0}^{L-1} \sum_{j=0}^t T_{ij}(i-j)}{\sum_{i=t+1}^{L-1} \sum_{j=0}^t T_{ij}}$$

soit $J(t) = \frac{a}{c} + \frac{b}{d}$ avec $a = \sum_{i=0}^t \sum_{j=t+1}^{L-1} T_{ij}(i-j)$ $b = \sum_{i=0}^{L-1} \sum_{j=0}^t T_{ij}(i-j)$ $c = \sum_{i=0}^t \sum_{j=t+1}^{L-1} T_{ij}$

$$d = \sum_{i=t+1}^{L-1} \sum_{j=0}^t T_{ij}$$

Le seuil optimal t^* est ainsi obtenu en maximisant $J(t)$.

D'autres méthodes de seuillage basées sur les matrices de cooccurrence telles que celles basées sur la distance d'Ali-silvey [Chang, 1994], [Liane, 1997], ont été proposées.

Comme dans le cas d'un histogramme monodimensionnel, la segmentation par seuillage à partir de plusieurs seuils d'un histogramme bidimensionnel représenté par une matrice de cooccurrence consiste à répartir les pixels en K classes (C_1, C_2, \dots, C_K) à partir d'un ensemble de seuils $T = \{t_1, t_2, \dots, t_{K-1}\}$. Par convenance, on utilise aussi deux autres seuils, $t_0 = 0$ et $t_k = L-1$. Un pixel de niveau de gris $l(x, y)$ est affecté à la classe C_k si $t_k \leq l(x, y) \leq t_{k+1}$ avec $k = 0, 1, 2, \dots, K-1$.

La matrice de cooccurrence est dans ce cas découpée en plusieurs régions comme le montre la figure (2.2). Les zones situées sur la diagonale correspondent aux classes alors que les zones hors diagonale correspondent aux transitions entre les classes ou au bruit.

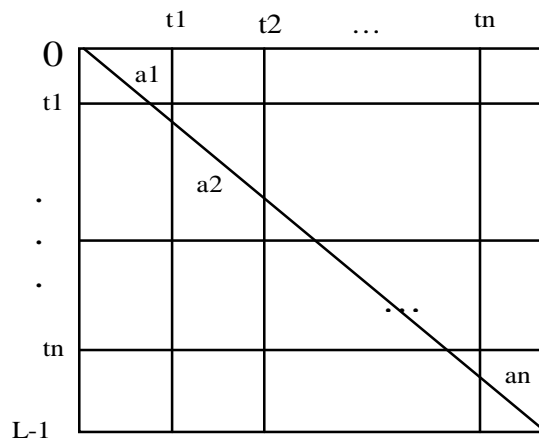


Fig. 2.2 : Seuillage d'une matrice cooccurrence à partir de plusieurs seuils

Les méthodes de calcul d'un seul seuil à partir des matrices de cooccurrence citées précédemment peuvent être facilement étendues pour le calcul de plusieurs seuils.

2.4. Seuillage d'histogrammes bidimensionnels

Comme pour une matrice de cooccurrence, l'histogramme bidimensionnel peut être réparti en quatre quadrants (Fig. 2.3.a) sauf que, il faut utiliser pour cela un vecteur seuils (s, t) . Les quadrants 1 et 2 correspondent respectivement à la distribution des pixels appartenant à l'objet et au fond, tandis que les quadrants 3 et 4 décrivent la distribution des pixels appartenant aux contours et au bruit. Les éléments proches de la diagonale correspondent aux pixels des zones homogènes de l'image alors que les éléments éloignés de la diagonale sont formés par les pixels de contour. La détermination du couple de seuils s et t à partir de l'histogramme bidimensionnel est généralement basée sur l'extension au cas bidimensionnel des méthodes de seuillage d'histogramme monodimensionnel comme celles présentées dans le chapitre 1.

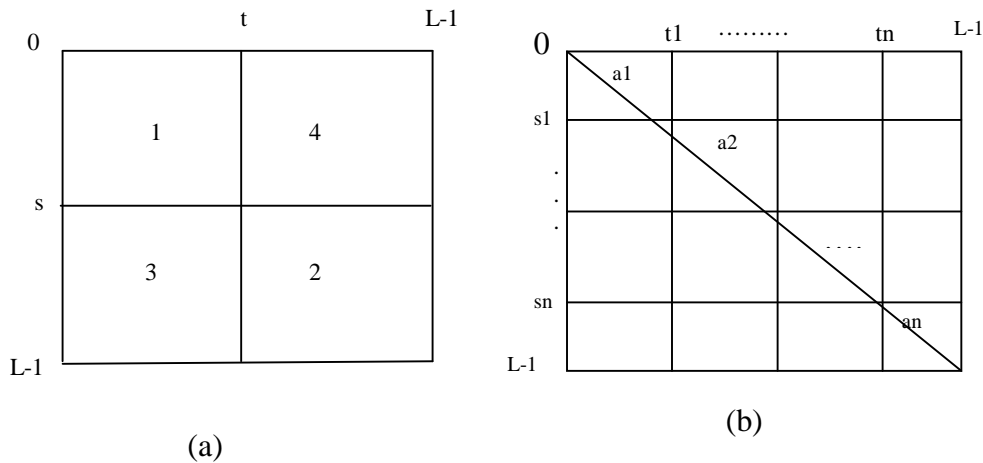


Fig. 2.3 : Histogrammes bidimensionnels (a) un seul seuil, (b) plusieurs seuils

Les méthodes de seuillage basées sur les histogrammes bidimensionnels peuvent être également étendues dans le cas du multi-seuillage. Il s'agira, donc, de déterminer un vecteur de k couples de seuils $[(s_1, t_1), (s_2, t_2), \dots, (s_{k+1}, t_{k+1})]$

Sachant que l'échelle des niveaux de gris s'étale de 0 à $L-1$, nous posons $(s_0, t_0) = (0, 0)$ et $(s_{k+1}, t_{k+1}) = (L-1, L-1)$. De plus, nous rappelons que le vecteur seuil contient une suite ordonnée croissante:

$$s_0 < s_1 < \dots < s_{k+1} \text{ et } t_0 < t_1 < \dots < t_{k+1}.$$

2.4.1. Seuillage d'Otsu bidimensionnel

Introduite par Liu et Li [Liu et al., 1993], cette méthode suppose que l'image est répartie en deux classes C_1 et C_2 (objet et fond). Les probabilités a priori des deux classes sont :

$$w_0 = P_r(C_1) = \sum_{i=0}^s \sum_{j=0}^t p_{ij} = w_0(s, t)$$

$$w_1 = P_r(C_2) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p_{ij} = w_1(s, t)$$

Les niveaux moyen des classes correspondantes sont donnés par :

$$\begin{aligned} \mathbf{m}_0 &= (m_{0i}, m_{0j})^T = \left(\sum_{i=0}^s iP_r\left(\frac{i}{C_1}\right), \sum_{j=0}^t jP_r\left(\frac{j}{C_1}\right) \right)^T = \left(\sum_{i=0}^s \sum_{j=0}^t ip_{ij} / w_0, \sum_{i=0}^s \sum_{j=0}^t jp_{ij} / w_0 \right)^T \\ &= \left(m_i(s, t) / w_0(s, t), m_j(s, t) / w_0(s, t) \right)^T \end{aligned}$$

$$\begin{aligned} \mathbf{m}_1 &= (m_{1i}, m_{1j})^T = \left(\sum_{i=s+1}^{L-1} iP_r\left(\frac{i}{C_2}\right), \sum_{j=t+1}^{L-1} jP_r\left(\frac{j}{C_2}\right) \right)^T = \left(\sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} ip_{ij} / w_1, \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} jp_{ij} / w_1 \right)^T \\ &= \left(m_i(S, T) / w_1(S, T), m_j(S, T) / w_1(S, T) \right)^T \end{aligned}$$

Le vecteur total de niveau moyen est donné par :

$$\mathbf{m}_T = (m_{Ti}, m_{Tj})^T = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ip_{ij}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} jp_{ij} \right)^T$$

La matrice de variance interclasse est définie par :

$$\mathbf{S}_B = \sum_{i=0}^L P_r(C_i) [(m_i - m_T)(m_i - m_T)^T]$$

Le vecteur seuil optimal (s^*, t^*) qui maximise la trace de la matrice $\mathbf{S}_B(s, t)$:

$$t_r \mathbf{S}_B(s^*, t^*) = \max_{I \leq s, T \leq I} \{t_r \mathbf{S}_B(s, t)\}$$

avec

$$t_r \mathbf{S}_B \approx \frac{[m_{Ti} w_0(s, t) - m_i(s, t)]^2 + [m_{Tj} w_0(s, t) - m_j(s, t)]^2}{w_0(s, t)[1 - w_0(s, t)]}$$

La recherche du couple de seuils (s, t) par une méthode exhaustive exige un temps de calcul assez élevé. En effet, l'évaluation de la fonction objective pour l'ensemble des couples (s, t)

nécessite au moins L^2 opérations. Pour accélérer les calculs, Gong et al. [Gong et al, 1998] ont proposé une procédure itérative qui consiste à déterminer les valeurs des seuils de la manière itérative suivante :

$$w_0(s,0) = w_0(s-1,0) + P_{s_0}$$

$$w_0(s,t) = w_0(s,t-1) + w_0(s-1,t) - w_0(s-1,t-1) + P_{st}$$

$$m_i(s,0) = m_i(s-1,0) + sP_{s_0}$$

$$m_i(s,t) = m_i(s,t-1) + m_i(s-1,t) - m_i(s-1,t-1) + sP_{st}$$

$$m_j(s,0) = m_j(s-1,0) + P_{st}$$

$$m_j(s,t) = m_j(s,t-1) + m_j(s-1,t) - m_j(s-1,t-1) + tP_{st}$$

La méthode d'Otsu, décrite ci-dessus, peut facilement être généralisée à la segmentation avec plusieurs seuils.

$$((s_1^*, t_1^*), \dots, (s_K^*, t_K^*)) = \text{Arg max}[t_r, \mathbf{S}_B]$$

$$\text{avec } t_r, \mathbf{S}_B = \frac{\sum_{k=1}^K (m_{T_k} w_k - m_k)^2}{\prod_{k=1}^K w_k}$$

2.4.2. Entropie de Shannon à deux dimensions

Abutaleb a été le premier à avoir appliqué le principe de l'entropie de Shannon 2D sur des histogrammes à deux dimensions [Abutaleb, 1989]. Les probabilités des classes C_1 (fond) et C_2 (objet) sont notées P_1 et P_2 . Pour un vecteur seuil (s,t) donné, P_1 et P_2 sont définies comme suit:

$$P_1 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} \quad P_2 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij}$$

Compte tenu de la définition de l'entropie 2D :

$$H = - \sum_i \sum_j p_{ij} \log(p_{ij})$$

les entropies des classes C_1 et C_2 s'écrivent :

$$H^1(s,t) = - \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \frac{p_{ij}}{P_1} \log\left(\frac{p_{ij}}{P_1}\right)$$

$$H^2(s, t) = - \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \frac{P_{ij}}{P_2} \log \left(\frac{P_{ij}}{P_2} \right)$$

L'entropie totale est alors calculée comme suit :

$$H^T(s, t) = H^1(s, t) + H^2(s, t)$$

D'après le principe du maximum d'entropie, le vecteur de seuillage optimal (s^*, t^*) est celui qui maximise l'entropie totale :

$$H^T(s^*, t^*) = \arg \max \{H^T(s, t)\}$$

Pour accélérer les calculs, Gong et al. [Gong et al., 1998] ont également modifié l'expression de la probabilité joint P_{st} par l'expression récurrente suivante :

$$P_{ST} = \sum_{i=0}^s \sum_{j=0}^{t-1} P_{ij} + \sum_{i=0}^s P_{iT} = P_{s(t-1)} + \sum_{i=0}^{s-1} P_{it} + P_{st} = P_{s(t-1)} + P_{(s-1)t} - P_{(s-1)(t-1)} + P_{st}$$

$$P_{s0} = P_{(s-1)0} + P_{s0}$$

Ainsi, on peut déterminer à chaque instant P_{st} d'une manière récurrente.

De la même manière, ils ont exprimé l'entropie a posteriori sous la forme suivante :

$$H_{s0} = H_{(s-1)0} - P_{s0} \ln P_{s0}$$

$$H_{st} = H_{s(t-1)} + H_{(s-1)t} - H_{(s-1)(t-1)} - P_{st} \ln P_{st}$$

Le principe du seuillage par maximisation de l'entropie 2D, décrit ci-dessus, peut facilement être généralisé à la segmentation avec plusieurs seuils. L'entropie totale de plusieurs classes peut être définie à partir de $(K-1)$ couples de seuils (s_i, t_i) , $i = 1, 2, \dots, K-1$

$$H^T((s_1, t_1), (s_2, t_2), \dots, (s_K, t_K)) = - \sum_{k=1}^K H(s_k, t_k)$$

$$\text{avec } H(s_k, t_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \frac{P_{ij}}{P_k} \log \frac{P_{ij}}{P_k}$$

$$\text{et } P_k = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} P_{ij}$$

2.4.3. Entropie exponentielle bidimensionnelle

L'entropie exponentielle 2D est définie dans [Nakib et al. 2007. 1, 2] par :

$$H_a = \left(\sum_i \sum_j p_{ij}^a \right)^{\frac{1}{1-a}}$$

Les probabilités des classes C_1 et C_2 sont notées P_1 et P_2 . Pour un vecteur seuils (s, t) donné, P_1 et P_2 sont définies comme suit:

$$P_1 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij}, \quad P_2 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij}$$

L'entropie exponentielle 2D des classes C_1 et C_2 sont données par :

$$H^{(1)} = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \left(\frac{p_{ij}}{P_1} \right)^a \right)^{\frac{1}{(1-a)}}, \quad H^{(2)} = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \left(\frac{p_{ij}}{P_2} \right)^a \right)^{\frac{1}{(1-a)}}$$

L'entropie 2D totale est :

$$H_a^T = H_a^{(1)} + H_a^{(2)}$$

En appliquant le principe du maximum de l'entropie, le vecteur de seuils optimaux (s^*, t^*) doit satisfaire :

$$H_a^T(s^*, t^*) = \max(H_a^T)$$

Dans le cas de la segmentation avec plusieurs seuils, le vecteur de seuils optimaux $(a_1^*, \dots, a_K^*) = ((s_1^*, t_1^*), \dots, (s_K^*, t_K^*))$ est obtenu en maximisant l'entropie exponentielle 2D suivante:

$$(a_1^*, \dots, a_K^*) = \arg \max (H_a^T(a_1, \dots, a_{K-1}))$$

$$\text{avec } t_0 < t_1 < \dots < t_{K-1} < L-1 \quad \text{et } s_0 < s_1 < \dots < s_{K-1} < L-1$$

$$H_a^T(a_1, \mathbf{K}, a_K) = \sum_{k=1}^K H_a^{(k)}$$

$$\text{avec } H_a^{(k)} = \left(\sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \left(\frac{p_{ij}}{P_k} \right)^a \right)^{\frac{1}{(1-a)}}$$

$$\text{avec } P_k = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} p_{ij}$$

2.4.4. Entropie de Renyi bidimensionnelle

L'entropie 2D de Renyi d'ordre a associée à l'image est définie [Prasanna et al., 2004] par :

$$H_T^a = \frac{1}{1-a} \log \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j)^a$$

Avec $a \neq 1$ et $a \in \mathfrak{R}$ est paramètre

$\lim_{a \rightarrow 1} H_T^a = H_T$ représente l'entropie de Shannon généralisée telle que :

$$H_T = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j) \log p(i, j)$$

Etant donné les densités $P_2(s, t)$ et $P_4(s, t)$ associées respectivement aux classes « objet » et « fond », nous avons $P_4(s, t) \approx 1 - P_2(s, t)$

$$\text{avec } P_2(s, t) = \sum_{i=0}^s \sum_{j=0}^t p(i, j) \text{ et } P_4(s, t) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p(i, j)$$

Les entropies 2D de Renyi d'ordre a associées aux classes « objet » et « fond » sont définies par :

$$H_b^a(s, t) = \frac{1}{1-a} \log \sum_{i=0}^s \sum_{j=0}^t \left(\frac{p(i, j)}{P_2(i, j)} \right)^a$$

$$H_w^a(s, t) = \frac{1}{1-a} \log \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \left(\frac{p(i, j)}{1 - P_2(i, j)} \right)^a$$

Nous avons supposé que les probabilités proches de la diagonale sont négligeables et $H_w^a(s, t)$ est calculé en utilisant $P_2(s, t)$ au lieu $P_4(s, t)$.

Le couple de seuils optimal (s, t) est déterminé en maximisant $H_w^a(s, t) + H_b^a(s, t)$

$$(s^*, t^*) = \text{Arg} \max_{(s, t) \in G \times G} [H_w^a(s, t) + H_b^a(s, t)]$$

Notons que les limites des entropies de Renyi associées aux classes « objet » et « fond » aboutissent aux entropies de Shannon.

$$\lim_{a \rightarrow 1} H_b^a(s, t) = \lim_{a \rightarrow 1} \frac{\frac{d}{da} \left[\log \sum_{i=0}^s \sum_{j=0}^t \left(\frac{p(i, j)}{P_2(s, t)} \right)^a \right]}{\frac{d}{da} (1-a)} = H_b(s, t)$$

$$\lim_{a \rightarrow 1} H_w^a(s, t) = \lim_{a \rightarrow 1} \frac{\frac{d}{da} \left[\log \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \left(\frac{p(i, j)}{1 - P_2(s, t)} \right)^a \right]}{\frac{d}{da} (1-a)} = H_w(s, t)$$

Par conséquent :

$$\lim_{a \rightarrow 1} [H_b^a(s, t) + H_w^a(s, t)] = H_b(t, s) + H_w(s, t)$$

L'entropie 2D de Renyi, décrite ci-dessus, peut facilement être généralisée dans le cas de plusieurs classes. Les seuils optimaux sont déterminés selon la formule suivante :

$$((s_1^*, t_1^*), \dots, (s_K^*, t_K^*)) = \text{Arg max} \left[\sum_{k=1}^K H_k^a(s_k, t_k) \right]$$

avec

$$H_k^a(s_k, t_k) = \frac{1}{1-a} \log \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \left(\frac{p(i, j)}{P_k(i, j)} \right)^a$$

$$P_k(s_k, t_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} p(i, j)$$

2.4.5. Entropie de Tsallis-Havrda-Charvat bidimensionnelle

Havrda et Charvat [Havrda et Charvat, 1976] ont définie une entropie de degré a par :

$$H^a = \frac{1}{1-2^{1-a}} \left[1 - \sum_{i=1}^n p_i^a \right]$$

Indépendamment, Tsallis [Tsallis, 1988] a proposé de généraliser l'entropie de Shannon comme suit :

$$H^a = \frac{1}{1-a} \left[1 - \sum_{i=1}^n p_i^a \right]$$

avec $a \neq 1$ un paramètre réel ($a \in \mathfrak{R}$)

Sahoo et al. ont proposé de combiner ces deux entropie dans le cadre du seuillage d'un histogramme bidimensionnel [Sahoo et al., 2006]. Ils définissent l'entropie bidimensionnelle de Tsallis-Havrda-Charvat associée à la classe « objet » par :

$$H_b^a = \frac{1}{a-1} \left[1 - \sum_{i=0}^s \sum_{j=0}^t \left(\frac{p(i, j)}{P_2(s, t)} \right)^a \right]$$

$$\text{avec } P_2(s, t) = \sum_{i=0}^s \sum_{j=0}^t p(i, j)$$

et l'entropie bidimensionnelle de Tsallis-Havrda-Charvat associée à la classe « fond » par :

$$H_w^a = \frac{1}{a-1} \left[1 - \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \left(\frac{p(i, j)}{P_4(s, t)} \right)^a \right]$$

$$\text{avec } P_4(s, t) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} p(i, j)$$

En approximant $P_4(s, t)$ par $(1 - P_2(s, t))$, on obtient :

$$H_w^a = \frac{1}{a-1} \left[1 - \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \left(\frac{p(i, j)}{1 - P_2(s, t)} \right)^a \right]$$

L'entropie totale de Tsallis-Havrda-Charvat est donnée par :

$$H_T^a = \frac{1}{a-1} \left[1 - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j)^a \right]$$

Le vecteur seuil optimal (s^*, t^*) est déterminé comme suit

$$(s^*, t^*) = \text{Arg} \max_{(s, t) \in G \times G} \Phi_a(s, t)$$

$$\text{Avec } \Phi_a(s, t) = H_b^a(s, t) + H_w^a(s, t) + (1-a)H_b^a(s, t)H_w^a(s, t)$$

L'entropie Tsallis-Havrda-Charvat 2D, décrite ci-dessus, peut facilement être généralisée pour le multi-seuillage. L'entropie totale de plusieurs classes peut être définie à partir de $(k-1)$ couples de seuils (s_i, t_i) , $i = 1, 2, \dots, k-1$ et les seuils optimaux sont déterminés en maximisant l'entropie totale :

$$((s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)) = \arg \max \left(\sum_{k=1}^K H_k^a(s_k, t_k) \right)$$

$$\text{avec } H_k^a(s_k, t_k) = \frac{1}{a-1} \left[1 - \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \left(\frac{p(i, j)}{P_k} \right)^a \right]$$

$$\text{et } P_k(s_k, t_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} p(i, j).$$

2.5. Histogramme à trois dimensions

L'histogramme monodimensionnel d'une image représente seulement la distribution de niveau gris de l'image, il ne prend pas en considération la corrélation spatiale entre les pixels de l'image. Pour améliorer la segmentation par seuillage d'histogrammes, l'utilisation des histogrammes bidimensionnels représentant la distribution conjointe des niveaux de gris et de niveaux de gris moyens ou les moyennes locale $g(x, y)$ est préconisée. Malheureusement, dans certaines situations (bruit important) et mauvais contraste, l'histogramme bidimensionnel ne possède pas de vallées bien distinctes ce que ne facilite pas la tâche de la recherche des seuils.

Récemment Jing et al [Jing et al., 2003] ont proposé d'utiliser un histogramme tridimensionnel en tenant en compte de la médiane du voisinage de chaque pixel de l'image. Ainsi chaque pixel de coordonnées (x, y) d'une image peut être représenté par un triplet $[l(x, y), g(x, y), h(x, y)]$.

A partir de l'ensemble des triplets, on peut définir un histogramme à trois dimension $L \times L \times L$. La valeur d'un élément (i, j, k) de l'histogramme 3D est dénoté par p_{ijk} , lequel représente la probabilité du vecteur (i, j, k) tel que :

$$p_{ij} = \frac{h_{ijk}}{N^2} \quad \text{avec } i, j, k = 0, 1, 2, \dots, L-1 \quad \text{tel que } i = l(x, y) \text{ et } j = g(x, y) \text{ et } k = h(x, y)$$

$$\sum_{i=1}^{L-1} \sum_{j=1}^{L-1} \sum_{k=1}^{L-1} p_{ijk} = 1$$

h_{ijk} représente la fréquence d'apparition de pixel, telle que

$$i = l(x, y)$$

$$g(x, y) = \frac{1}{k^2} \sum_{i=-\frac{k}{2}}^{\frac{k}{2}} \sum_{j=-\frac{k}{2}}^{\frac{k}{2}} l(x+i, y+j)$$

$$h(x, y) = \text{med}\{f(x+i, y+j)\}$$

$$i = -k/2, \mathbf{L}, k/2; j = -k/2, \mathbf{L}, k/2$$

On définit un vecteur de seuils tridimensionnel (s, t, v) tel que $0 \leq s, t, v \leq L-1$. La fonction de seuillage tridimensionnel est définie par :

$$b_{stv}(x, y) = \begin{cases} C_1 & \text{si } l(x, y) < s \text{ et } g(x, y) < t \text{ et } h(x, y) < v \\ C_2 & \text{si } l(x, y) \geq s \text{ ou } g(x, y) \geq t \text{ ou } h(x, y) \geq v \end{cases}$$

avec $0 \leq s, t, v \leq L-1$.

L'histogramme tridimensionnel peut être divisé en huit régions rectangulaires par l'intermédiaire du triplet seuils (s, t, v) , voir (Fig.2.9). Les régions 0 et 1 contiennent respectivement la distribution des pixels appartenant à l'objet et au fond, tandis que les régions de 2 à 7 contiennent la densité des pixels appartenant aux contours et au bruit.

Les éléments proches de la diagonale correspondent aux pixels des zones homogènes de l'image alors que les éléments éloignés de la diagonale sont formés par les pixels de contour. La détermination des seuils s, t, v à partir de l'histogramme tridimensionnel est généralement

basée sur l'extension au cas tridimensionnel des méthodes monodimensionnelles et bidimensionnelles.

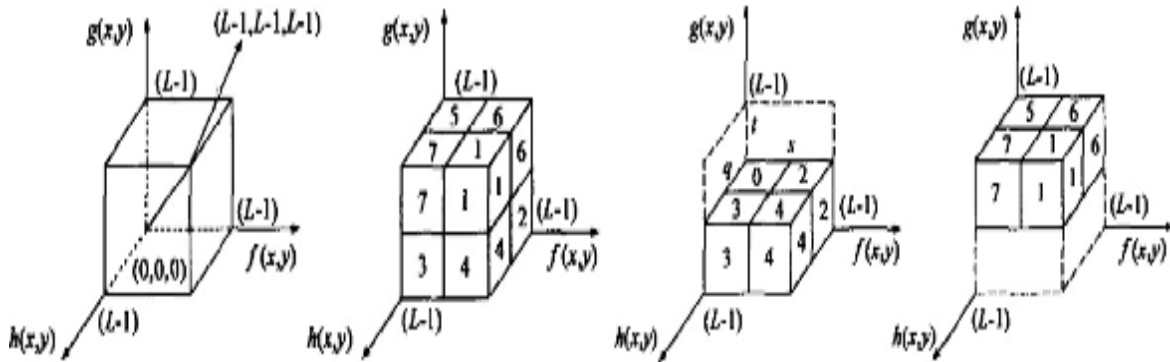


Fig.2.9. Histogramme a trois dimensions

2.5.1. Méthode d'Otsu tridimensionnelle

Cette méthode consiste à répartir les pixels l'image en deux classes C_1 et C_2 (objet et fond), par un vecteur seuil tridimensionnel [Jing et al, 2003]. Les densités de probabilité des pixels dans les régions 2 à 7 sont négligées lors de calcul du triplet (t, s, v) et seuls les densités de probabilités des régions 0 et 1 sont prises en compte. Soient w_0 et w_1 , les probabilités à priori des deux classes :

$$w_0 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \sum_{k=0}^{v-1} p_{ijk} = w_0(s, t, v)$$

$$w_1 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \sum_{k=v}^{L-1} p_{ijk} = w_1(s, t, v)$$

Les vecteurs moyen correspondants à la classe « fond » et à la classe « objet » sont données respectivement par :

$$m_0 = (m_{0i}, m_{0j}, m_{0k})' = \left[\frac{\sum_{(i,j,k) \in C_0} i \cdot p_{ijk}}{w_0}, \frac{\sum_{(i,j,k) \in C_0} ij p_{ijk}}{w_0}, \frac{\sum_{(i,j,k) \in C_0} k \cdot p_{ijk}}{w_0} \right]'$$

$$m_1 = (m_{1i}, m_{1j}, m_{1k})' = \left[\frac{\sum_{(i,j,k) \in C_1} i \cdot p_{ijk}}{w_1}, \frac{\sum_{(i,j,k) \in C_1} ij p_{ijk}}{w_1}, \frac{\sum_{(i,j,k) \in C_1} k \cdot p_{ijk}}{w_1} \right]'$$

et la moyenne totale de l'espace tridimensionnel est donnée par :

$$\mathbf{m}_T = (\mathbf{m}_{Ti}, \mathbf{m}_{Tj}, \mathbf{m}_{Tk})' = \left[\sum_{i,j,k} i \cdot p_{ijk}, \sum_{i,j,k} j \cdot p_{ijk}, \sum_{i,j,k} k \cdot p_{ijk} \right]'$$

Sachant que les densités de probabilité des régions de 2 à 7 sont négligées, on a :

$$w_0 + w_1 \approx 1$$

$$\mathbf{m}_T \approx w_0 \times \mathbf{m}_0 + w_1 \times \mathbf{m}_1$$

La matrice de déviation interclasses peut être définie comme suit

$$\mathbf{S}^2(s, t, v) = w_0 [(\mathbf{m}_0 - \mathbf{m}_T)(\mathbf{m}_0 - \mathbf{m}_T)^T] + w_1 [(\mathbf{m}_1 - \mathbf{m}_T)(\mathbf{m}_1 - \mathbf{m}_T)^T]$$

Le vecteur seuil optimal (s^*, t^*, v^*) est donné par :

$$(s^*, t^*, v^*) = \arg \max_{I \leq s, t, v \leq I} \{t_r[\mathbf{S}_B(s, t, v)]\}$$

avec

$$t_r[\mathbf{S}_B(s, t, v)] = \frac{[(w_0 \times \mathbf{m}_{Ti} - \mathbf{m}_{0i})^2 + (w_0 \times \mathbf{m}_{Tj} - \mathbf{m}_{0j})^2 + (w_0 \times \mathbf{m}_{Tk} - \mathbf{m}_{0k})^2]}{[w_0 \times (1 - w_0)]}$$

où $t_r[\mathbf{S}_B(s, t, v)]$ représente la trace de la matrice $\mathbf{S}_B(s, t, v)$

La méthode d'Otsu, décrite ci-dessus, peut facilement être généralisée dans le cas de multi-seuillage :

$$((s_1^*, t_1^*, v_1^*), \dots, (s_K^*, t_K^*, v_K^*)) = \text{Arg max} [t_r \mathbf{S}_B(s_k, t_k, v_k)]$$

2.5.2. Entropie de Kapur à trois dimensions

Les densités de probabilités des régions 0 et 1 sont notées P_1 et P_2 . Pour un vecteur (t, s, v) donné, P_1 et P_2 sont définies comme suit:

$$P_1 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \sum_{k=0}^{v-1} p_{ijk} \quad P_2 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \sum_{k=v}^{L-1} p_{ijk}$$

Nous définissons les entropies de la région 0 et la région 1 par :

$$H^1(s, t, v) = - \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \sum_{k=0}^{v-1} \frac{p_{ijk}}{P_1} \log \frac{p_{ijk}}{P_1}$$

$$H^2(s, t, v) = - \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \sum_{k=v}^{L-1} \frac{p_{ijk}}{P_2} \log \frac{p_{ijk}}{P_2}$$

L'entropie totale est alors calculée comme suit :

$$H^T(s, t, v) = H^1(s, t, v) + H^2(s, t, v)$$

D'après le principe du maximum d'entropie, le vecteur de seuil optimal (s^*, t^*, v^*) est celui qui maximise l'entropie totale :

$$H^T(s^*, t^*, v^*) = \text{Max}\{H^T(s, t, v)\}$$

Le principe du seuillage basé sur l'entropie de Kapur 3D peut être facilement généralisé au multi-seuillage :

$$H^T((s^*_1, t^*_1, v^*_1), (s^*_2, t^*_2, v^*_2), \dots, (s^*_k, t^*_k, v^*_k)) = \max\left(-\sum_{k=1}^K H(s_k, t_k, v_k)\right).$$

$$H(s_k, t_k, v_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \sum_{l=v_{k-1}}^{v_k-1} \frac{P_{ijl}}{P_k} \log \frac{P_{ijl}}{P_k}$$

$$\text{avec } P_k = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \sum_{l=v_{k-1}}^{v_k-1} P_{ijl}$$

2.5.3. Entropie exponentielle tridimensionnelle

On peut définir l'entropie exponentielle tridimensionnelle par :

$$H_a = \left(\sum_i \sum_j \sum_k P_{ijk}^a \right)^{\frac{1}{1-a}}$$

a est paramètre réel $a \in \mathfrak{R}$ $a \neq 1$ et $a \in \mathfrak{R}$

L'entropie exponentielle tridimensionnelle 3D de la classe (C_1) et de la classe (C_2) sont alors :

$$H^{(0)} = \left(\sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \sum_{k=0}^{v-1} \left(\frac{P_{ijk}}{P_1} \right)^a \right)^{\frac{1}{(1-a)}}$$

$$H^{(1)} = \left(\sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \sum_{k=v}^{L-1} \left(\frac{P_{ijk}}{P_2} \right)^a \right)^{\frac{1}{(1-a)}}$$

Le triplet de seuils optimaux doit satisfaire :

$$H_a^T(s^*, t^*, v^*) = \max(H_a^T(s, t, v))$$

$$\text{avec } H_a^T(s^*, t^*, v^*) = H_a^{(0)} + H_a^{(1)}$$

L'entropie exponentielle 3D, décrite ci-dessus, peut facilement être généralisée au multi-seuillage. En appliquant le principe du maximum de l'entropie, le vecteur de seuils optimaux $(a_1^*, \dots, a_k^*) = ((s_1^*, t_1^*, v_1^*), \dots, (s_k^*, t_k^*, v_k^*))$ doit satisfaire :

$$H_a^T(a_1^*, \dots, a_k^*) = \max(H_a^T(a_1, \dots, a_{k-1}))$$

$$\text{où } H_a^T(a_1^*, \dots, a_k^*) = \sum_{k=1}^K H^{(k)}$$

$$\text{avec } H^{(k)} = \left(\sum_{i=s_{k-1}}^{s_n-1} \sum_{j=t_{k-1}}^{t_n-1} \sum_{l=v_{k-1}}^{v_k-1} \left(\frac{P_{ijl}}{P_k} \right)^a \right)^{\frac{1}{(1-a)}} \text{ et } P_k = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{s_k-1} \sum_{l=v_{k-1}}^{v_k-1} p_{ijl}$$

2.5.4. Entropie de Renyi tridimensionnelle

On peut aussi définir l'entropie de Renyi d'ordre a associée à l'image par :

$$H_T^a = \frac{1}{1-a} \log \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p(i, j, k)^a \text{ telle que } \lim_{a \rightarrow 1} H_T^a = H_T$$

avec a est un paramètre réel $a \in \mathfrak{R}, a \neq 1$

$$H_T = - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p(i, j, k) \log p(i, j, k) \text{ représente l'entropie de Shannon.}$$

Soient $P_2(s, t, v)$ et $P_4(s, t, v)$ les densités associées aux classes « objet » et « fond » respectivement.

Nous posons $P_4(s, t, v) \approx 1 - P_2(s, t, v)$

$$\text{avec } P_2(s, t, v) = \sum_{i=0}^s \sum_{j=0}^t \sum_{k=0}^v p(i, j, k) \text{ et } P_4(s, t, v) = \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=v+1}^{L-1} p(i, j, k)$$

Les entropies de Renyi d'ordre a associées aux deux classes sont définies par :

$$H_b^a(s, t, v) = \frac{1}{1-a} \log \sum_{i=0}^s \sum_{j=0}^t \sum_{k=0}^v \left(\frac{p(i, j, k)}{P_2(i, j, k)} \right)^a$$

et

$$H_w^a(s, t, v) = \frac{1}{1-a} \log \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=v+1}^{L-1} \left(\frac{p(i, j, k)}{1 - P_2(i, j, k)} \right)^a$$

En négligeant les probabilités proches de la diagonale, $H_w^a(s, t, v)$ peut être calculée en utilisant $P_2(s, t, v)$ au lieu $P_4(s, t, v)$. Le triplet de seuils optimal (s, t, v) est obtenu en maximisant $H_w^a(s, t, v) + H_b^a(s, t, v)$

$$(s^*, t^*, v^*) = \text{Arg} \max_{(s, t, v) \in L \times L \times L} [H_w^a(s, t, v) + H_b^a(s, t, v)]$$

L'entropie Renyi 3D, décrite ci-dessus, peut facilement être généralisée pour le multi-seuillage :

$$\left((s_1^*, t_1^*, v_1^*), \dots, (s_k^*, t_k^*, v_k^*) \right) = \text{Arg} \max \left[\sum_{k=1}^K H_k^a(s_k, t_k, v_k) \right]$$

avec

$$H_k^a(s_k, t_k, v_k) = \frac{1}{1-a} \log \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \sum_{l=v_{k-1}}^{v_k-1} \left(\frac{p(i, j, l)}{P_k(i, j, l)} \right)^a$$

$$\text{et } P_k(s_k, t_k, v_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{s_k-1} \sum_{l=v_{k-1}}^{v_k-1} p(i, j, l).$$

2.5.5. Entropie de Tsallis-Havrda-Charvat tridimensionnelle

Comme dans le cas mono et bidimensionnel, nous définissons l'entropie de Tsallis-Havrda-Charvat associées aux classes « objet » et « fond » par :

$$H_b^a = \frac{1}{a-1} \left[1 - \sum_{i=0}^s \sum_{j=0}^t \sum_{k=0}^v \left(\frac{p(i, j, k)}{P_2(s, t, v)} \right)^a \right]$$

$$H_w^a = \frac{1}{a-1} \left[1 - \sum_{i=s+1}^{L-1} \sum_{j=t+1}^{L-1} \sum_{k=v+1}^{L-1} \left(\frac{p(i, j, k)}{1 - P_2(s, t, v)} \right)^a \right]$$

L'entropie totale de Tsallis-Havrda-Charvat est alors :

$$H_T^a = \frac{1}{a-1} \left[1 - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} p(i, j, k)^a \right]$$

avec le paramètre $a \in \mathfrak{R}$, ($a \neq 1$)

Le triplet de seuils optimal peut être obtenu en appliquant le principe du maximum de l'entropie :

$$(s^*, t^*, v^*) = \text{Arg} \max_{(s, t, v) \in G^3} \Phi_a(s, t, v)$$

$$\text{avec } \Phi_a(S, T, V) = H_b^a(s, t, v) + H_w^a(s, t, v) + (1-a)H_b^a(s, t, v)H_w^a(s, t, v)$$

Dans le cas du multi-seuillage, l'ensemble des triplets de seuils peuvent être déterminé comme suit :

$$((s_1^*, t_1^*), \dots, (s_k^*, t_k^*)) = \text{Arg} \max \Phi(S, T, V)$$

$$\text{avec } \Phi_a(S, T, V) = \sum_{k=1}^K H_k^a(s_k, t_k, v_k) + (1-a) \prod_{k=1}^K H_k^a(s_k, t_k, v_k)$$

et

$$H_k^a(s_k, t_k, v_k) = \frac{1}{a-1} \left[1 - \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{t_k-1} \sum_{l=v_{k-1}}^{v_k-1} \left(\frac{p(i, j, l)}{P_k} \right)^a \right]$$

$$\text{avec } P_k(s_k, t_k, v_k) = \sum_{i=s_{k-1}}^{s_k-1} \sum_{j=t_{k-1}}^{s_k-1} \sum_{l=v_{k-1}}^{v_k-1} p(i, j, l).$$

2.6. Conclusion

Nous avons présenté dans ce chapitre quelques méthodes de seuillage d'histogrammes bidimensionnels et tridimensionnels. Ces méthodes ne sont que des extensions des méthodes de seuillage d'histogrammes monodimensionnels. Ces extensions sont simples puisqu'elles consistent à rajouter à chaque fois une dimension.

Notons que, dans la littérature du domaine du seuillage, seule la méthode d'Otsu à été proposée dans le cas de l'histogramme 3D. Nous avons ainsi proposé dans ce chapitre d'étendre les entropies de Kapur, exponentielle de Reny et Tsallis-Havrda-Charvat pour le seuillage d'histogrammes tridimensionnels. Nous avons également généralisé toutes les méthodes de seuillage des histogrammes bi et tridimensionnels au cas du multi-seuillage.

Chapitre 3

Etat de l'art sur les méthodes métaheuristiques

3.1 Introduction

La majorité des problèmes de seuillage d'image peuvent s'exprimer comme des problèmes d'optimisation qui peuvent être NP-difficiles et ne pourront pas être résolus de manière exacte dans un temps "raisonnable" puisque la capacité de calcul des machines évolue linéairement alors que le temps nécessaire à la résolution de ces problèmes évolue exponentiellement. Lorsqu'on s'attaque à des problèmes réels, il faut se résoudre à un compromis entre la qualité des solutions obtenues et le temps de calcul utilisé. Au milieu des années 1970 sont apparues des méthodes qui supervisent l'évolution de solutions fournies par des heuristiques. Ces méthodes assurent un compromis entre diversification (concentrer la recherche sur de mauvaises zones de l'espace de recherche) et intensification (rechercher les meilleures solutions dans la région de l'espace de recherche en cours d'analyse). Ces algorithmes ont été appelés "métaheuristiques" et ont pour objectif de trouver des solutions dont la qualité est au-delà de ce qu'il aurait été possible de réaliser avec une simple heuristique.

Dans ce chapitre, nous introduisons les différentes méthodes classiques d'optimisation métaheuristiques tout en rappelant celles qui ont été appliquées pour la segmentation d'images par seuillage d'histogrammes.

3.2 Définitions des notions d'heuristiques et de métaheuristiques

3.2.1. Heuristiques et Métaheuristiques

Une heuristique d'optimisation est une méthode approchée se voulant simple, rapide et adaptée à un problème donné. Sa capacité à optimiser un problème avec un minimum d'informations est contrebalancée par le fait qu'elle n'offre aucune garantie quant à l'optimalité de la meilleure solution trouvée. Du point de vue de la recherche opérationnelle, ce défaut n'est pas toujours un problème, tout spécialement quand seule une approximation de la solution optimale est recherchée.

Parmi les heuristiques, certaines sont adaptables à un grand nombre de problèmes différents sans changements majeurs dans l'algorithme, on parle alors de métaheuristiques. La plupart des heuristiques et des métaheuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire. En plus de cette base stochastique, les métaheuristiques sont généralement itératives, c'est-à-dire qu'un même schéma de recherche est appliqué plusieurs fois au cours de l'optimisation, et directes, n'utilisent pas l'information du gradient de la fonction objectif. Elles

tirent en particulier leur intérêt de leur capacité à éviter les optima locaux, soit en acceptant une dégradation de la fonction objectif au cours de leur progression, soit en utilisant une population de points comme méthode de recherche (se démarquant ainsi des heuristiques de descente locale). Souvent inspirées d'analogies avec la réalité (physique, biologie, éthologie, ...), elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent faire l'objet d'adaptations pour des problèmes continus.

3.2.2. Notion de voisinage

Plusieurs métaheuristiques font appel à la notion de voisinage. Rappelons que le problème d'optimisation consiste à trouver une solution x^* qui minimise une fonction objective f sur l'espace des solutions S . Un ensemble $N(x)$, appelé voisinage de x , est obtenu en effectuant des modifications locales sur x , nommées mouvements, en suivant des règles bien précises qui dépendent du problème considéré. A partir d'une solution initiale $x_0 \in S$ (fournie aléatoirement ou par une heuristique constructive), une méthode de recherche locale (de voisinage) génère des solutions x_1, x_2, \dots, x_n dans S telles que $x_{i+1} \in N(x_i)$. Les différents algorithmes de recherche locale se distinguent essentiellement par la manière de construire ou générer le voisinage et aussi par la manière de choisir une solution dans le voisinage $N(x)$ de la solution courante x . Le voisinage d'une solution est un sous-ensemble de solutions qu'il est possible d'atteindre par une série de transformations données. On peut ainsi définir le voisinage comme suit [Talbi et al., 2009] :

$$N(x) = \{y \in S : dist(x, y) \leq e\}.$$

3.3. Méthodes d'optimisation métaheuristiques

On peut diviser les méthodes d'optimisation métaheuristiques en deux grandes classes :

- § Les métaheuristiques à solution unique
- § Les métaheuristiques à solution multiple ou population de solution

La figure (3.1) suivante présente une taxinomie de quelques méthodes métaheuristiques existantes.

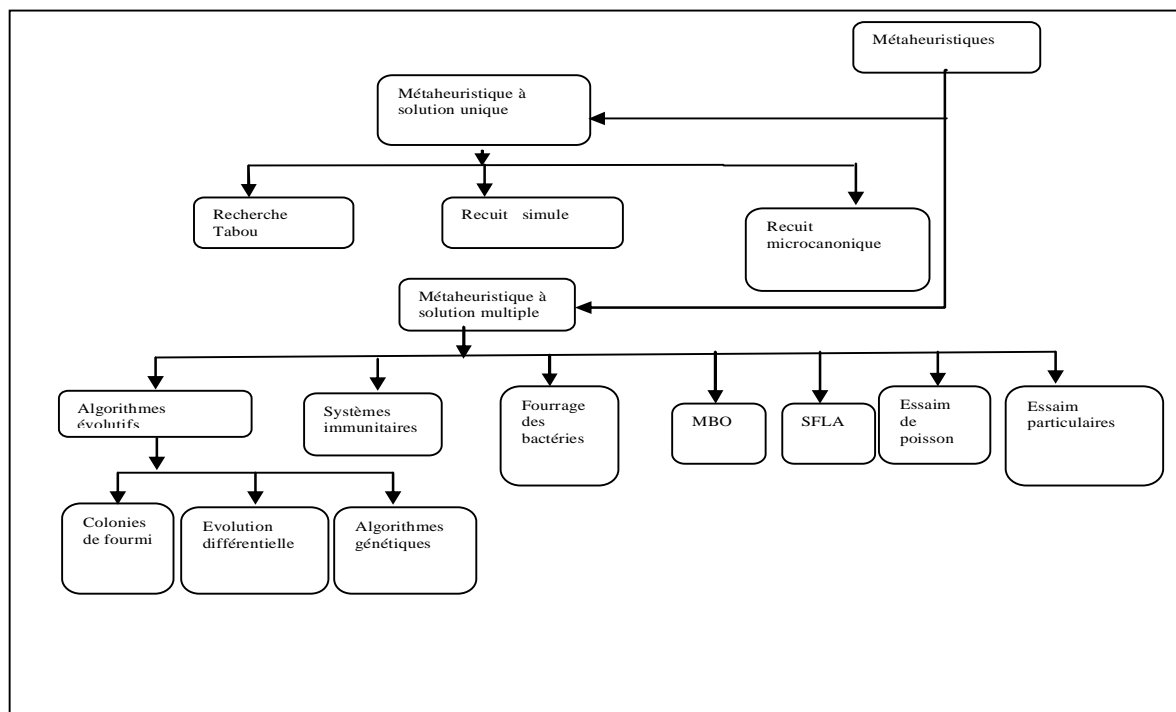


Fig. 3.1 : Taxinomie des méthodes d'optimisation métaheuristiques

3.3.1. Les métaheuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [Bachelet, 1999].

De manière générale, les opérateurs de recherche locale s'arrêtent quand une solution localement optimale est trouvée, c'est-à-dire quand il n'existe pas de meilleure solution dans le voisinage. Mais accepter uniquement ce type de solution n'est bien sûr pas satisfaisant. Dans un cadre plus général, il serait alors intéressant de pouvoir s'échapper de ces minima locaux (Fig. 3.2). Il faut alors permettre à l'opérateur de recherche locale de faire des mouvements pour lesquels la nouvelle solution retenue sera de qualité moindre que la précédente. Parmi les méthodes métaheuristiques à solution unique, on trouve, le recuit simulé, le recuit microcanonique et la recherche tabou.

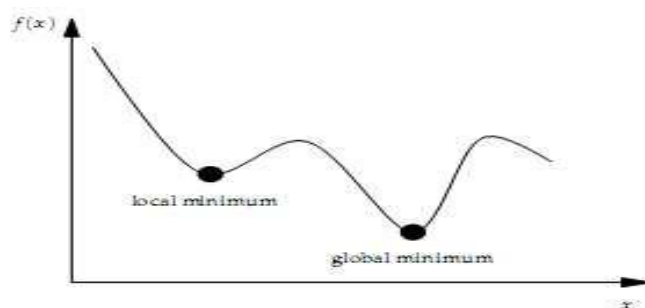


Fig.3.2. Minimum local et global

3.3.1.1. Le recuit simulé (Simulated Annealing)

Le recuit simulé est une technique d'optimisation, proposée initialement par Kirkpatrick [Kirkpatrick, 1983]. Elle s'inspire des méthodes de simulation de Métropolis (années 50) en mécanique statistique dont le principe est résumé dans l'algorithme 3.1 ci-dessous.

Initialiser un point de départ x_0 et une température T
Pour $i = 1$ à n **faire**
 Tant que x_i n'est pas accepté **faire**
 Si $f(x_i) \leq f(x_{i-1})$ accepté x_i
 Si $f(x_i) > f(x_{i-1})$ accepté x_i avec la probabilité $e^{-\frac{f(x_i) - f(x_{i-1})}{T}}$
 Fin tant que
Fin pour

Algorithme 3.1: Algorithme de Métropolis

L'analogie historique s'inspire du recuit des métaux en métallurgie. Un métal refroidi trop vite présente de nombreux défauts microscopiques, c'est l'équivalent d'un optimum local pour un problème d'optimisation combinatoire. Si on le refroidit lentement, les atomes se réarrangent, les défauts disparaissent et le métal a alors une structure très ordonnée équivalente à un optimum global. La méthode du recuit simulé, appliquée aux problèmes d'optimisation, considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. L'originalité de cette méthode est qu'il est possible de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle. Ceci permet d'échapper aux optima locaux. Au début de l'algorithme voir (Algorithme 3.2), un paramètre T , apparenté à la température, est déterminé et décroît tout au long de l'algorithme pour tendre vers 0. De la valeur de ce paramètre va dépendre la probabilité p d'acceptation des solutions détériorant (plus la température T est élevée, plus cette probabilité sera forte).

La performance du recuit simulé dépend, entre autres, de la règle de refroidissement (c'est à dire la décroissance du paramètre T) que l'on utilise. Un refroidissement trop rapide mènerait vers un optimum local pouvant être de très mauvaise qualité. Un refroidissement trop lent serait très coûteux en temps de calcul.

1. Fixer une solution aléatoire S
2. Calculer l'énergie $f(S)$ associée à cette solution S
3. Initialiser la température (T)
4. Tant que la condition d'arrêt n'est pas satisfaite faire
 - 4.1. Tant que l'équilibre thermodynamique n'est pas atteint faire
 - 4.1.1. Tirer une nouvelle solution S'
 - 4.1.2. Appliquer la règle de Métropolie
 - 4.1.3. Si $f(S') < f(S)$

$$S = S'$$

$$f(S) = f(S')$$
 Fin Si
 - 4.2. Décroître la température T
5. Afficher la solution optimale

Algorithme 3.2: Algorithme de recuit simulé

Le recuit simulé a connu un grand succès dans différents domaines d'application [Chen et al, 1999]. D'un point de vue théorique, le recuit simulé permet d'approcher de près la solution optimale du problème plus rapidement qu'une exploration exhaustive dans l'espace de recherche. En pratique, un bon réglage des paramètres internes de l'algorithme permet d'accélérer la convergence vers une solution pseudo optimale, avec une précision prédéfinie. Le recuit simulé a été aussi utilisé pour résoudre le problème du seuillage par l'approche paramétrique [Synder et al., 1990] et pour optimiser la fonction d'Otsu ou de Kapur [Campbell, 1996], [Hammouche et al., 2010], [Nakib, 2007.1].

3.3.1.2. Le recuit microcanonique

Cette méthode d'optimisation exploite des principes très proches de ceux du recuit simulé. Le recuit microcanonique effectue des paliers d'énergie totale décroissante en diminuant l'énergie cinétique entre deux paliers. Ainsi l'algorithme converge par réduction des énergies d'un ensemble de solutions autour de celles qui sont optimales [Ray et al, 1997. Woods et al, 1998]. Le recuit microcanonique met en œuvre l'algorithme de Creutz [Creutz, 1983] qui est basé sur l'évaluation d'une succession de transitions pour maximiser l'entropie pour une énergie totale constante. Cette énergie est fixée préalablement.

Pour une énergie totale E_t , l'algorithme itératif de Creutz permet de converger vers l'équilibre thermodynamique en répétant un grand nombre de fois les deux étapes suivantes :

- ∅ évaluation de la variation d'énergie associée à une transition élémentaire aléatoire de l'état courant i , d'énergie potentielle E_i , vers un nouvel état j , d'énergie E_j : $\Delta E_{ij} = E_j - E_i$.
- ∅ acceptation de cette transition vers ce nouvel état si $\Delta E_{ij} \leq E_i - E_i$.

Dans l'article original de Creutz [Creutz, 1983], le terme $E_i - E_i = D$ est appelé "Demon" et peut être interprété comme l'énergie cinétique du système à l'état i . Les transitions vers des états d'énergie potentielle plus élevée ne sont autorisées que s'il y a suffisamment d'énergie cinétique pour compenser l'augmentation d'énergie potentielle, et donc rester à énergie constante.

L'algorithme du recuit microcanonique consiste à réduire l'énergie totale, à partir d'une énergie totale élevée, au sein de l'algorithme de Creutz. Plusieurs lois de décroissance de l'énergie peuvent être utilisées, comme dans le cas du recuit simulé. Les différentes étapes de l'algorithme sont résumées dans (Algorithme 3.3). L'algorithme de Creutz est beaucoup plus simple que celui de Metropolis, et nécessite bien moins de calculs. Comme il n'y a pas besoin d'un générateur de nombres aléatoires, sa mise en œuvre est simplifiée par rapport à d'autres algorithmes. En comparaison avec le recuit simulé, dans le cas des problèmes de grande taille, plusieurs études ont montré que les résultats obtenus sont très proches, [Herault et al, 1993]. Dans certains cas, le recuit microcanonique peut être moins performant, à cause des barrières d'énergie infranchissables qu'il construit au cours de sa recherche et qui peuvent le piéger dans un minimum local.

1. Initialisation,
 - 1.1. Définir le pourcentage minimal p de transitions acceptées dans le premier palier
 - 1.2. Fixer E_i initial tel que le taux p des transformations testées soient acceptées.
 - 1.3. Tirer aléatoirement une solution acceptable et calculer son énergie E
 - 1.4. choisir le nombre maximal de transitions testées à chaque palier de l'énergie totale
2. Tant que le nombre de solutions acceptées est non nul faire
 - 2.1. Répéter pour un nombre d'itérations donnée, tirées aléatoirement une transition et calculer la variation d'énergie associée ΔE
 - 2.1.1. Accepter la transition si $\Delta E = E_i - E$
 - 2.1.2. Si $\Delta E \leq 0$, comparer le nouvel état au meilleur état trouvé depuis le début de la recherche, et le mémoriser s'il est meilleur.
 - 2.2. Décroître E_i : $E_i = aE_i$
- Fin tant que
3. Afficher le meilleur état rencontré au cours de la recherche

Algorithme 3.3: Algorithme du recuit microcanonique

L'utilisation du recuit microcanonique permet d'avoir de bons résultats et une reproductibilité meilleure, du fait de l'absence de caractère stochastique dans la recherche de la solution optimale. Le seul caractère probabiliste est celui de la recherche dans le voisinage. Du point de vue vitesse de convergence, cet algorithme est plus rapide que le recuit simulé, ce qui est en accord avec la comparaison faite dans [Herault, et al; 1993]. Dans [Nakib et al., 2007.1.2.4], les auteurs ont appliqué cet algorithme afin de résoudre le problème du seuillage par maximisation de l'entropie de Shannon à deux dimensions et l'entropie exponentielle.

3.3.1.3. La recherche Tabou

La technique Tabou (Tabu search) est une méthode itérative générale d'optimisation combinatoire qui a été élaborée par F. Glover [Glover, 1986]. Elle fait explicitement usage d'un transfert de connaissance entre des étapes successives de la recherche en introduisant la notion de la liste tabou. La liste taboue est une mémoire flexible qui garde une certaine trace des dernières opérations passées. On peut y stocker des informations pertinentes à certaines étapes de la recherche pour en profiter ultérieurement. Cette liste peut être statique (de taille fixe) ou dynamique (de taille variable) et permet d'empêcher les blocages dans les minima locaux en interdisant de passer à nouveau sur des configurations de l'espace de recherche précédemment visitées (problème de cycle). On peut distinguer deux types de listes :

- liste tabou explicite, elle sauvegarde des solutions complètes.
- § liste tabou attributive, elle sauvegarde seulement les informations à propos des solutions qui changent à travers la recherche.

La recherche taboue se montre très performante sur un nombre considérable de problèmes d'optimisation combinatoire. Les idées de base de la recherche Tabou se retrouvent également dans le travail de Hansen [Hansen, 1986]. Elle n'a aucun caractère stochastique et utilise la notion de mémoire pour éviter de tomber dans un optimum local.

Le principe de l'algorithme est le suivant, à chaque itération, le voisinage (complet ou sous-ensemble de voisinage) de la solution courante est examiné et la meilleure solution est sélectionnée. En appliquant ce principe, la méthode autorise de remonter vers des solutions qui semblent moins intéressantes mais qui ont peut être un meilleur voisinage. Le risque de cycler entre deux solutions est présent. Pour éviter ce problème, la méthode interdit de revenir vers une solution récemment visitée. Pour cela, une liste taboue contenant les attributs des dernières solutions visitées est tenue à jour. Chaque nouvelle solution considérée enlève de cette liste la solution la plus anciennement visitée. Ainsi, la recherche de la solution courante suivante se fait dans le voisinage de la solution courante actuelle sans considérer les solutions

appartenant à la liste tabou. Cette méthode ne s'arrête pas d'elle-même et il faut déterminer un critère d'arrêt en fonction du temps de recherche que l'on s'octroie. Ce critère peut être, par exemple, l'exécution d'un certain nombre d'itérations ou la non-amélioration de la meilleure solution pendant un certain nombre d'itérations. Ainsi, tout au long de l'algorithme, la meilleure solution doit être conservée car l'arrêt se fait rarement sur la meilleure solution

Il existe de nombreuses variantes de recherches Tabou impliquant des techniques plus ou moins sophistiquées pour intensifier ou diversifier la recherche. L'algorithme (3.4) ci-dessous montre comment minimiser une fonction $F(T)$ par la recherche tabou. Une application de cet algorithme pour déterminer un ensemble de seuils est donnée dans [Hammouche et al., 2010].

```

1) Générer aléatoirement le vecteur solution initial  $T$ 
2) Evaluer le vecteur et stocker la meilleure solution  $T^* (T^* = T)$ 
3) Initialiser la mémoire tabou  $Liste = f$ 
4) Pour un nombre d'itérations fixe
    " Créer le vecteur des meilleurs solution  $V_{best}$  "
    Construire un ensemble de solutions de voisinage  $V(T)$  autour du vecteur de solution  $T$ 
    Pour chaque  $V \in V(T)$ 
        Si  $V \notin Liste$  et  $F(V) > F(V_{best})$  alors  $V_{best} = V$ 
    Fin pour
    " Mise à jour de la mémoire tabou "
     $Liste = Liste \cup V_{best}$ 
    " Mise à jour de la meilleur solution "
    Si  $F(T^*) < F(V_{best})$  alors  $T^* = V_{best}$ 
5) Production de la meilleur solution enregistré  $T^*$ 

```

Algorithme 3.4: Méthode générique de Tabou

3.3.2. Les métaheuristiques à population de solutions

Ces méthodes d'optimisation sont basées sur une population de solutions qu'il faut améliorer au fur et à mesure des itérations. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité. Ces méthodes sont : évolutives et sont basés sur le principe du processus d'évolution naturelle analogue aux mécanismes d'évolution des espèces vivantes [Jong, 1993], [Bäck, 1993a], [Schoenauer, 1997]. Un algorithme évolutif typique est composé de trois éléments essentiels :

- § une **population** constituée de plusieurs individus représentant des solutions potentielles (configurations) du problème donné;
- § un **mécanisme d'évaluation** de l'adaptation de chaque individu de la population à l'égard de son environnement extérieur;

§ un **mécanisme d'évolution** composé d'opérateurs permettant d'éliminer certains individus et de produire de nouveaux individus à partir des individus sélectionnés.

Du point de vue opérationnel, un algorithme évolutif typique débute avec une population initiale souvent générée aléatoirement et répète, ensuite, un cycle d'évolution composé de 3 étapes séquentielles :

- a. Mesurer l'adaptation (la qualité) de chaque individu de la population par le mécanisme d'évaluation,
- b. Sélectionner une partie des individus,
- c. Produire de nouveaux individus par des recombinaisons d'individus sélectionnés.

Ce processus se termine quand la condition d'arrêt est vérifiée, qui se traduit, par exemple, par le nombre maximum de cycles (générations) ou un nombre maximum d'évaluations à atteindre.

Selon l'analogie de l'évolution naturelle, la qualité des individus de la population devrait tendre à s'améliorer au fur et à mesure du processus. Parmi les composantes d'un algorithme évolutif, l'individu et la fonction d'adaptation correspondent respectivement à la notion de configuration et à la fonction d'évaluation dans les méthodes de voisinage. La notion de mécanisme d'évolution est proche de celle du mécanisme de parcours du voisinage; en particulier, on peut interpréter une itération d'une méthode de voisinage comme un mécanisme d'évolution portant sur une population réduite à un seul individu. Cependant, les opérateurs sont sensiblement différents. En effet, un algorithme évolutif comporte un ensemble d'opérateurs tels que la sélection, la mutation et éventuellement le croisement.

Plusieurs algorithmes métaheuristiques d'optimisation ont été proposés. Nous présenterons dans ce qui suit certains d'entre eux, ceux qui sont utilisés dans le seuillage.

3.3.2.1. Les algorithmes génétiques (Genetic Algorithms)

Les algorithmes génétiques sont des méthodes basées sur les mécanismes biologiques tels que les lois de Mendel et sur le principe fondamental (sélection) de Charles Darwin. Holland exposa les principes de ces algorithmes pour permettre aux ordinateurs "d'imiter les êtres vivants " en évoluant pour rechercher la solution à un problème [Holland, 1975]. Il expliqua d'abord comment ajouter de l'intelligence dans un programme informatique avec les croisements (échange du matériel génétique) et la mutation (source de la diversité génétique).

Les algorithmes génétiques simulent le processus d'évolution d'une population. A partir d'une population $P(t)$ de solutions du problème représentant des individus, on applique des opérateurs simulant les interventions sur le génome tel que le croisement (cross-over) ou la mutation pour arriver à une population de solutions de mieux en mieux adaptée au problème. Cette adaptation est évaluée grâce à une fonction coût. L'algorithme (3.5) montre le principe de l'algorithme génétique

1. Initialisation de $P(t)$
2. Evaluer chaque individu de $P(t)$
3. Tant que le critère d'arrêt n'est pas satisfait faire
 - 3.1. $t = t + 1$
 - 3.2. Sélectionner $P(t+1)$ à partir de $P(t)$
 - 3.3. Croisement $P(t+1)$
 - 3.4. Muter $P(t+1)$
 - 3.5. Evaluer $P(t+1)$
- Fin tant que
4. Afficher le meilleur état rencontré au cours de la recherche

Algorithme 3.5 : Principe d'un algorithme génétique.

La procédure de sélection consiste à favoriser la propagation des meilleures solutions parmi la population, tout en préservant la diversité génétique, afin d'explorer de nouvelles régions de l'espace de recherche. Plusieurs opérateurs de sélection ont été proposés dans la littérature [Dréo et al., 2003]. Le but de la procédure de croisement est de maintenir la diversité en manipulant les composantes des individus (chromosomes). De nouveaux individus, appelés enfants, sont créés en échangeant des parties des individus de la population (parents). Cet échange se fait en sélectionnant un point de découpage, aléatoirement (avec une densité de probabilité uniforme), et en échangeant ensuite les deux sous chaînes de chacun des deux parents. La procédure de mutation consiste à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire. La mutation est l'opérateur qui apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace de recherche. Les AGs ont été intensivement utilisés pour le seuillage d'histogramme [Hammouche et al., 2008], on peut citer également les méthodes basées sur la minimisation de l'entropie floue [Yang et al., 2003], l'entropie relative [Tao et al., 2003] et l'optimisation des fonctions d'Otsu et de Kapur [Hammouche et al., 2010], ainsi que celle utilisées pour le seuillage d'histogrammes bidimensionnels [Wang et al., 2009], [Yu and Fan, 2008] .

3.3.2.2. Algorithmes de l'évolution différentielle (differential evolution)

Proposé par Prince et Storn [Price and Storn, 1995], l'algorithme d'évolution différentielle (DE) est une version améliorée de l'AG. Il dispose d'opérateurs de mutation, croisement et sélection. Cependant l'algorithme DE marque la différence avec l'AG en donnant plus d'importance à la mutation. Cette mutation a pour but de créer une solution « mutant » à partir des différences entre deux solutions de la population. Plusieurs stratégies ont été alors proposées [Price and Storn, 1995], elles diffèrent par le choix de la solution à muter, le nombre de différences, le type de croisement. La stratégie la plus classique dite DE/rand/1/exp consiste à ajouter à une solution m , choisie aléatoirement dans la population, une différence pondérée de deux autres solutions n et l prises aléatoirement dans la population avec $m \neq n \neq l$:

$$y = X_m + F(X_n - X_l)$$

F étant un facteur.

Le croisement consiste à changer chaque gène de la solution j par celui du mutant avec une probabilité Cr. La nouvelle solution C ainsi créée est sélectionnée pour remplacer une solution de la population si sa fitness est meilleure. Ce mécanisme de mutation, croisement et sélection est répété pour chaque solution de la population (Algorithme 3.6).

```

1) Initialiser la population X de solutions
2) Evaluer chaque solution de la population X de solutions
3) Fixer le nombre d'itérations.
Pour tout les individus j de la population X
"Création le vecteur de différence "
  Générer aléatoirement l, m, n avec j ≠ m ≠ n ≠ l.
"Calculer le vecteur solution Y "
  y_k = X_m + F(X_n - X_l)
"Croisement "
"Générer un vecteur de solution C "
  Si rand(0,1) < c_k alors c_k = y_k sinon c_k = x_jk (k = 1, K, N)
"Evaluer le vecteur solution C"
" Sélection"
  Si F(C) < F(X_j)
    X_j = C
    F(X_j) = F(C)
Fin pour // j
Fin pour // itération
4) Production de la meilleure solution enregistré T* de la population Pop

```

Algorithme 3.6 : Algorithme de l'évolution différentielle

L'algorithme d'évolution différentielle est appliqué dans le cadre du seuillage en optimisant la fonction d'Otsu et [Hammouche et al., 2010] et pour l'approximer l'histogramme par mélange de distributions [Cuevas et al., 2010].

3.3.2.3. Les colonies de fourmis (Ants System)

L'optimisation à base des colonies des fourmis proposée par Dorigo [Dorigo et al., 1996] consiste à mimer le comportement des fourmis lors de la recherche de nourriture. En se déplaçant du nid à la source de nourriture et vice-versa (ce qui, dans un premier temps, se fait essentiellement d'une façon aléatoire), les fourmis déposent au passage sur le sol une substance odorante appelée phéromone, ce qui a pour effet de créer une piste chimique. Les fourmis peuvent sentir ces phéromones qui ont un rôle de marqueur de chemin. Quand les fourmis choisissent leur chemin, elles ont tendance à choisir la piste qui porte la plus forte concentration de phéromones (voir figure 3.3). Cela leur permet de retrouver le chemin vers leur nid lors du retour. D'autre part, les odeurs peuvent être utilisées par d'autres fourmis pour retrouver les sources de nourriture détectées par leurs consœurs. Il a été démontré expérimentalement que ce comportement permet l'émergence des chemins les plus courts entre le nid et la nourriture, à condition que les pistes de phéromones soient utilisées par une colonie entière de fourmis.

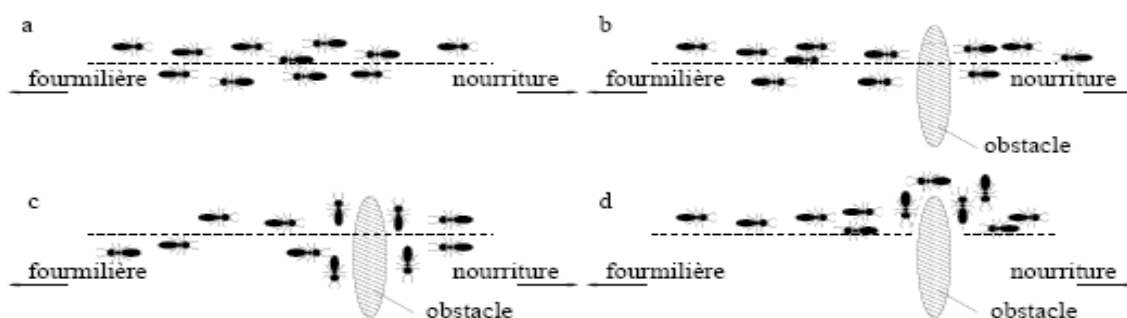


Fig. 3.3. (a) Les fourmis suivent un chemin entre la fourmilière et la nourriture. (b) Un obstacle apparaît sur le chemin ; les fourmis choisissent entre prendre à droite et à gauche avec équiprobabilité. (c) La phéromone s'évapore sur le chemin le plus long. (d) Toutes les fourmis choisissent le chemin le plus court.

La figure (3.4) illustre le principe de base d'un algorithme d'optimisation par les colonies de fourmis

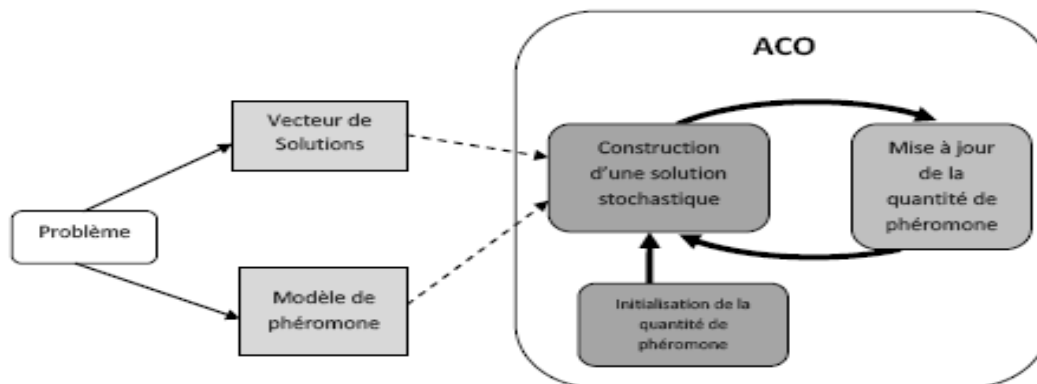


Fig. 3.4 : Principe de base de l'algorithme ACO.

L'ACO a été utilisé dans plusieurs domaines [Diaf et al., 2009]. Son utilisation dans le seuillage d'histogramme est illustré par l'algorithme (3.7) [Hammouche et al., 2010]. Il consiste à générer une population de solution (chaque solution représente une fourmi artificielle) à partir d'une matrice de phéromones notée t où un élément t_{ik} de cette matrice représente la trace de phéromone qui lie le niveau de gris i au $k^{\text{ème}}$ seuil. Ainsi le niveau de gris pour lequel t_{ik} est max est choisi comme le seuil t_k ($t_k = \arg \max_{i \in [0, L-1]} t_{ik}$).

```

1) Création de la matrice de phéromone  $t$ 
2) Initialiser la population de solutions  $X$ 
3) Déterminer la meilleure solution  $T^*$  de la population
4) Pour le nombre d'itération fixe
   " Détermination de la population Pop avec la matrice des phéromones  $Z$ "
     Pour tous individus  $j$  de la population  $pop$ 
       Si  $rand > q_0$  alors
          $pop(j)_{-t_k} = \arg \max_{i \in [g_{\min}, g_{\max}]} t_{ik}$ 
       Si non  $pop(j)_{-t_k} = rand_{INT} [g_{\min}, g_{\max}]$ 
     Evaluer la population  $pop$  "évaluation toute les solutions candidates"
   "Mise à jours les meilleures solutions  $T^*$   $T^*$ "
   Comparer les meilleurs individus  $T$  de  $POP$  avec  $T^*$ 
     Si  $T$  à une fitness meilleure que  $T^*$ , alors remplaçant  $T^*$  par  $T$ 
      $F_{\max} = F(T^*)$ 
   Mise à jour le chemin des phéromones
   Fin pour // itération
      $t_{ik} = r t_{ik} + (1 - r) F_{\max}$ 
5) Production de la meilleure solution  $T^*$  et l'enregistrée.

```

Algorithme 3.7: Algorithme de colonie de fourmis

3.3.2.4. Systèmes immunitaires artificiels (Artificial Immune Systems)

Les méthodes d'optimisation basées sur systèmes immunitaires artificiels (AIS) sont apparues dans les années 90 [Leandro et al., 2002]. Ils s'inspirent du fonctionnement du système immunitaire humain qui est doté d'un mécanisme de défense contre les agressions d'organismes extérieurs et de la capacité d'apprendre. La métaphore dont sont issus les algorithmes AIS met l'accent sur les aspects d'apprentissage et de mémoire du système immunitaire dit adaptatif (par opposition au système dit inné), notamment via la discrimination entre le soi et le non-soi. En effet, les cellules vivantes disposent sur leurs membranes de molécules spécifiques dites "antigènes". Chaque organisme dispose ainsi d'une identité unique, déterminée par l'ensemble des antigènes présents sur ses cellules. Les lymphocytes (un type de globule blanc) sont des cellules du système immunitaire qui possèdent des récepteurs capables de se lier spécifiquement à un antigène unique, permettant ainsi de reconnaître une cellule étrangère à l'organisme. Un lymphocyte ayant ainsi reconnu une cellule du non-soi va être stimulée à proliférer (en produisant des clones de lui-même) et à se différencier en cellules permettant de garder en mémoire l'antigène, ou en cellules permettant de combattre les agressions. Dans le premier cas, il sera capable de réagir plus rapidement à une nouvelle exposition à l'antigène, c'est le principe même de l'efficacité des vaccins. Dans le second cas, le combat contre les agressions est possible grâce à la production d'anticorps. La figure 3.4 résume ces principales étapes. Il faut également noter que la diversité des récepteurs dans l'ensemble de la population des lymphocytes est, quant à elle, produite par un mécanisme d'hyper-mutation des cellules clonées. Les principales idées utilisées pour la conception de la métaheuristique sont les sélections opérées sur les lymphocytes, accompagnées par les rétroactions positives permettant la multiplication et la mémoire du système. En effet, ces caractéristiques sont capitales pour maintenir les caractéristiques auto-organisées du système. On peut, dans le cadre de l'optimisation, considérer les AIS comme une forme d'algorithme évolutionnaire présentant des opérateurs particuliers. Pour opérer la sélection, on se fonde par exemple sur une mesure d'affinité (i.e. entre le récepteur d'un lymphocyte et un antigène), la mutation s'opère quant à elle via un opérateur d'hyper-mutation directement issu de la métaphore. Au final, l'algorithme obtenu est très proche d'un algorithme génétique (voir algorithme 3.8).

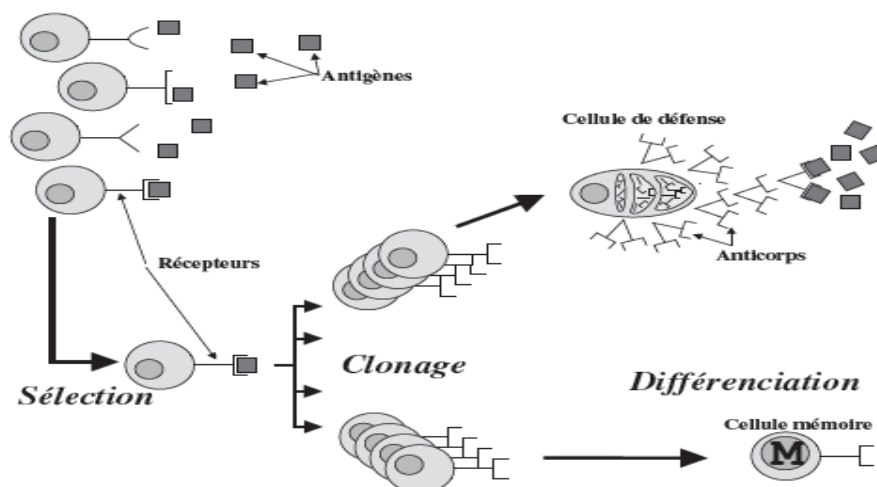


Fig. 3.4. La sélection par clonage des lymphocytes, présentant des récepteurs spécifiques d'un antigène, se différencie en cellule mémoire ou en cellule participant à la défense active de l'organisme par les anticorps.

1. Engendrer un ensemble de solutions P , composée d'un ensemble de cellules mémoires P_m ajoutées à la population présente P_r , $P = P_m + P_r$
2. Déterminer les n meilleures cellules P_n parmi la population P
3. Imiter les n individus pour former une population C . Le nombre de clones produits pour chaque cellule est fonction de l'affinité
4. Effectuer une hyper-mutation des clones, engendrer ainsi une population C . La mutation est proportionnelle à l'affinité
5. Sélectionner les m individus de C pour former la population mémoire P_m
6. Remplacer les plus mauvais individus dans P pour former P_r
7. Si un critère d'arrêt n'est pas atteint, retourner en 1.

Algorithme 3.8: Algorithme de système immunitaire artificiel

Les systèmes immunitaires artificiels ont été appliqués dans différents domaines tels que la robotique, la détection d'anomalies [De Castro and Von Zuben, 2000]. Ils ont été également appliqués dans le cadre du seuillage pour déterminer les paramètres statistiques des distributions gaussiennes qui composent l'histogramme [Cuevas et al, 2009].

3.3.2.5. Optimisation par colonie d'abeille

L'optimisation par Colonie d'Abeilles (BSO : Bee Swarm Optimization) est une famille très récente des métaheuristiques dont le principe est basé sur le comportement des abeilles [Abbass, 2001]. Elle se présente sous forme de deux variantes principales: l'optimisation par mariage d'abeilles (MBO: Marriage Bee Optimization), basé sur les phénomènes d'évolution des abeilles qui font intervenir parfois des méthodes de voisinage et l'optimisation par la danse d'abeilles (DBO : Danse Bée Optimization) qui est basé sur le comportement des

abeilles lors de la recherche de nourriture. Les abeilles lorsqu'elles cherchent leur nourriture démarrent d'un point particulier, à partir de ce point, elles s'éloignent de certaines distances dans différentes directions, elles commencent ensuite leurs recherches. De retour au nid, elles produisent une danse qui indiquera à ses congénères la qualité de la meilleure source de nourriture qu'elles aient trouvée. La danse permet à elle seule de comparer dans l'absolu les sources de nourriture visitées. La source la plus riche deviendra le nouveau point de recherche. L'abeille qui a découvert une source de nourriture revient à la ruche et communique non seulement la direction de la nourriture mais également la distance entre la ruche et l'alimentation ainsi que la qualité de la nourriture trouvée. L'abeille transmet ces informations par une danse. Il s'agit d'une véritable communication, d'un véritable langage codé, les abeilles informées pourraient trouver la nourriture ainsi signalée.

L'algorithme MBO prend en compte les communautés d'abeilles qui comprennent trois types d'individus anatomiquement différents : La Reine (femelle), la mère de tous les individus qui la composent : les faux bourdons, les ouvrières et les futures reines. Sa capacité à pondre est très importante. La reine s'accouple généralement avec cinq ou six mâles, les faux-bourdons (mâles) et les ouvrières (femelles stériles) sont toujours beaucoup plus nombreuses que les mâles. Elles sont incapables de s'accoupler et donc de se reproduire. Elles sécrètent la cire, construisent les alvéoles (creux), récoltent le nectar, le pollen (poussière produite par les étamines des plantes à fleurs et constituée de grains contenant les cellules reproductrices mâles) et l'eau, transforment le nectar (liquide sucré sécrété par les plantes et que viennent butiner les insectes) en miel, éclosent les œufs, nettoient la ruche et si nécessaire, la défendent contre les prédateurs. La seule fonction du mâle des abeilles (faux-bourdon) est de s'accoupler avec les nouvelles reines. Après l'accouplement, qui a lieu en vol, le mâle meurt rapidement. Chaque couche sociale a un rôle à jouer dans la naissance d'une nouvelle génération d'abeilles et de subvenir aux besoins de la colonie. Lorsque la reine s'accouple avec les faux bourdons dans les airs pendant un vol appelé vol nuptial, le sperme du faux-bourdon est ajouté à la spermathèque de la reine (réservoir de spermatozoïdes). La reine se servira toute sa vie de cette réserve pour féconder les œufs. Un ovule fécondé donnera naissance à une abeille femelle, soit reine, soit ouvrière, et un ovule non fécondé donnera naissance à un faux-bourdon.

Dans l'algorithme MBO, une reine artificielle possède essentiellement un génotype, une spermathèque, une vitesse et une énergie. Le génotype de la reine peut être considéré comme une solution du problème traité. Les nouvelles solutions ne sont autres que les œufs de la reine. Le processus de reproduction tend à améliorer le génotype de la reine au fil des

génération et donc d'améliorer la solution initiale. Les Ouvrières artificielles sont des heuristiques parfois spécifiques au problème traité dont le rôle est d'améliorer les génotypes des couvées (futurs reines), et donc, permettre d'obtenir de meilleures solutions. Un faux-bourdon est constitué d'un génotype complet et d'un masque servant à masquer la moitié des gènes (choisis aléatoirement). Par contre, Les faux-bourdons réels ne possèdent que la moitié d'un génotype. La moitié non masquée du génotype constitue le sperme du faux-bourdon. Lors de la fécondation, les gènes du mâle (éléments de son génotype) sont croisés avec le génotype de la reine pour former un nouveau génotype qui constitue une solution complète du problème.

L'algorithme MBO est présenté ci-dessus, au début de l'algorithme, cinq paramètres sont fixés: (1) nombre de reines, (2) nombre d'ouvrières, (3) nombre de couvées, (4) nombre de vols nuptiaux et (5) taille de la spermathèque de la reine. Dans le processus de recherche, les reines représentent des solutions, tandis que les ouvrières représentent l'heuristique employée pour la recherche locale (amélioration).

Au début de l'algorithme, les ouvrières sont initialisées avec certaines heuristiques. Un ensemble de reines s'est produit aléatoirement et leurs génotypes se sont améliorés en utilisant une heuristique (ouvrière) pour préserver seulement les meilleures reines. Un ensemble de vols nuptiaux sont alors entrepris sachant que la vitesse et l'énergie de chaque reine est initialisé aléatoirement à une valeur comprise entre [0.5, 1] pour s'assurer qu'il volera pendant un certain nombre de fois. Les transitions effectuées par chaque reine sont en fonction de cette vitesse et énergie. La vitesse de la reine représente la probabilité d'inverser chaque bit du génome du faux-bourdon. À chaque itération d'un vol, la reine s'accouple avec un faux-bourdon rencontré durant sa trajectoire selon une règle de probabilité. Si l'accouplement est réussi, le sperme du faux-bourdon (génotype) est ajouté à la spermathèque de la reine.

Quand toutes les reines ont terminé leurs vols, le procédé de création de couvée commence. Pour créer une nouvelle couvée, une reine est choisie selon le coût de son génotype et le sperme est choisi aléatoirement à partir de la spermathèque de la reine, ensuite, on croise le sperme avec le génotype de la reine. La mutation est alors appliquée à la nouvelle couvée pour l'améliorer, en utilisant les ouvrières. Les nouvelles couvées améliorées sont alors triées selon leur forme physique (fitness) et elles remplacent les reines de mauvaises qualités jusqu'à ce qu'il n'y ait aucune couvée meilleure que n'importe quelle reine.

Les couvées restantes sont alors détruites et un nouveau vol nuptial est entrepris. Ceci est répété jusqu'à ce que tous les vols nuptiaux soient générés ou un critère d'arrêt soit vérifié.

L'algorithme 3.9: suivant illustre le principe de la méthode de mariage d'abeilles (MBO: Marriage in Honey Bees Optimization,)

```

- Initialiser la taille  $M$  du spermathèque
- Initialiser  $E(t)$  et  $V(t)$ : Energie et vitesse des reines respectivement
- Initialiser les génotypes des reines aléatoirement
Pour un nombre prédéfini de vols nuptiaux faire
    Tant Que (les critères d'arrêt ne sont pas satisfaits) Faire
        Pour chaque reine
             $t = 0$ ; Initialiser  $E(t)$  et  $V(t)$ : à des valeurs aléatoires entre  $[0.5,1]$ 
             $step = (0.5 * E(t)) / M$  (Facteur de réduction de l'énergie)
            Générer un faux-bourdon aléatoirement ;
            Tant Que ( $E(t) > 0$ ) faire
                - Générer un nouveau Faux-bourdon en changeant chaque bit du génotype en utilisant la vitesse
                  comme probabilité
                - Evaluer la fitness du nouveau faux-bourdon
                Si (le nouveau faux-bourdon est meilleur que l'ancien) Alors
                    Si (le faux-bourdon passe la règle de probabilité) Alors
                        Si (la spermathèque de la reine n'est pas pleine) Alors
                            Ajouter le sperme du faux-bourdon a la spermatique de la reine
                        Fin Si
                    Fin Si
                -  $t = t + 1$ ;  $E(t) = E(t) - step$ ;  $V(t) = 0.9 * V(t)$ 
                - Flipper chaque bit de Génotype de Faux-bourdon avec la probabilité de la vitesse
                Fin Si
            Fin Tant Que
        Pour  $i=1$  jusqu'à nombre total de couvées Faire
            - Sélectionner un sperme aléatoirement dans la spermatique
            - Générer une couvée en utilisant le croisement et la mutation avec le sperme sélectionne
            - Amélioration des œufs par les ouvrières
            - Mettre a jour le fitness des ouvrières
        Fin Pour
    Fin Pour

```

Algorithme 3.9: Algorithme MBO

On trouve dans [Horng, 2009] et [Azorbad et al, 2011] l'utilisation de l'algorithme MBO dans le seuillage des histogrammes d'image infrarouge.

3.3.2.6. Algorithme par sauts de grenouilles

Cette algorithme connu sous son appellation anglophone "Shuffled Frog-Leaping algorithm" (SFLA) est une méthode métaheuristique d'optimisation qui imite l'évolution d'un groupe de grenouilles en cherchant un endroit qui dispose d'une quantité maximale de nourriture étalée aléatoirement sur des pierres dans un étang [Eusuff et Lanssy, 2000].

Dans la méthode SFLA, les grenouilles peuvent communiquer entre elles et améliorer leurs mêmes par contamination (passant l'information). Chaque grenouille, représente une

solution au problème. La population entière des grenouilles est divisée à un certain nombre de groupe de grenouilles appelées communautés qui évoluent indépendamment pour parcourir l'espace des solutions dans les différentes directions. Chaque communauté représente un type de même conduisent localement à l'exploration de l'espace des solutions selon des stratégies spécifiques qui permet le transfert des mêmes parmi les individus locaux.

L'information entre les différentes communautés circule par l'intermédiaire d'un processus de saut.

Supposons que la population initiale est constituée de F grenouilles définie aléatoirement dans l'espace $(X_i \quad i=1,2,\dots,F)$. L'adaptabilité ou la fitness $f(i)$ de la $i^{\text{ème}}$ grenouille représente la valeur de la fonction objective. Toutes les grenouilles sont triées dans un ordre décroissant et sont divisées en m communautés contenant chacune n grenouilles ($F = m \times n$), de telle manière que la première grenouille appartient à la première communauté, la deuxième grenouille est affectée à la deuxième communauté etc.

Dans chaque communauté, les grenouilles fournissant la meilleure et la plus mauvaise solution sont identifiées et notée par X_b et X_w respectivement. La grenouille donnant la meilleure solution dans la population entière est notée par X_g

Pendant l'évolution d'une communauté, c.-à-d., pendant l'exploration locale, la plus mauvaise grenouille effectue un saut vers la meilleure X_b selon la règle suivante:

$$S = r \times (X_b - X_w)$$

$$X'_w = X_w + S. \quad (S < S_{\max})$$

Où S représente la valeur du saut, r est un nombre aléatoire compris entre $(0,1)$, S_{\max} est le saut maximal autorisé.

Si le saut produit une meilleure solution, alors cette solution X'_n remplace la plus mauvaise X_n . Sinon, on applique la même règle en remplaçant cette fois-ci X_b par la solution globale X_g . Si la nouvelle solution X'_n reste moins bonne que X_w , on génère alors aléatoirement une solution X_n meilleure que celle de X_w et on remplace X_w par X'_w . Chaque communauté évolue ainsi en un certain nombre d'itérations.

Les m communautés sont ensuite mélangés pour former à nouveau la population et le processus de tri, partitionnement de la population en communautés, la recherche locale et le mélange des communautés est répété jusqu'à satisfaire un critère d'arrêt.

L'algorithme ci-dessous décrit le principe de SFLA :

Étape 1 : Fixer la taille F de la population, le nombre m de communauté et le nombre N d'itérations

Étape 2 : Générer aléatoirement une population de F solutions et évaluer chaque solution

Étape 3 : Trier la population et déterminer la meilleure solution X_g

Étape 4 : Partitionner la population en m communautés

Étape 5 : "Recherche local"

Pour chaque communauté

Répéter pendant N itérations

- Déterminer la meilleure solution X_b et la mauvaise solution X_w
- Calculer la solution X'_w à partir de X_b
- Si X'_w est meilleure que X_w , remplacer X_w par X'_w
- Sinon calculer X'_w à partir de X_g

Remplacer X_w par X'_w

- Si X'_w est meilleure que X_w
- Sinon générer aléatoirement X'_w et remplacer X_w

Fin si

Fin pour

Étape 6 : Regrouper les m communautés pour former à nouveau la population

Étape 7 : Aller à l'étape 3 si le critère d'arrêt n'est pas atteint

Algorithme 3.10: Pseudo code de SFLA

[Na Wang et al, 2010] ont proposé une méthode de seuillage d'un histogramme à base du SFLA. La fonction d'Otsu à trois dimensions est utilisée comme fonction objective.

3.3.2.7. L'optimisation par essais particuliers

La métaheuristique basée sur la méthode des essais particuliers ("Particle Swarm Optimization", PSO) a été développée par Kennedy et al. en 1995 [Kennedy, et al, 1995]. Le principe de la méthode provient analogiquement avec les comportements collectifs d'animaux comme le déplacement des bancs de poissons ou le vol des oiseaux. En effet, on peut observer chez ces animaux des dynamiques de déplacement relativement complexes, alors qu'individuellement chaque individu a une intelligence limitée et une connaissance seulement locale de sa situation dans l'essaim. Un individu de l'essaim n'a pour connaissance que la position et la vitesse de ses plus proches voisins. Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement. Des règles simples, telles que, aller à la même vitesse que les autres, se déplacer dans la même direction, ou encore rester proche de ses voisins : sont des exemples de comportements qui suffisent à maintenir la cohésion de l'essaim, et qui permettent la mise en œuvre de comportements collectifs complexes et adaptatifs. L'intelligence globale de l'essaim est donc la conséquence directe des interactions locales

entre les différentes particules de l'essaim. La performance du système entier est supérieure à la somme des performances de ses parties.

Kennedy et Eberhart se sont inspirés de ces comportements socio-psychologiques pour créer le PSO. Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, survole, l'espace de recherche, en quête de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes :

- § Une composante physique : la particule tend à suivre sa direction courante de déplacement ;
- § Une composante cognitive : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;
- § Une composante sociale : la particule tend se diriger vers le meilleur site déjà atteint par ses voisins. Dans le cas d'un problème d'optimisation, la qualité d'un site de l'espace de recherche est déterminée par la valeur de la fonction objective en ce point.

La Fig.3.5 illustre le schéma vectorielle du déplacement d'une particule qui détermine de nouvelle position d'une particule dans un processus PSO (les trois flèches grisées représentent la combinaison vectorielle des trois effets pris en compte) [Ou et Lin, 2006]

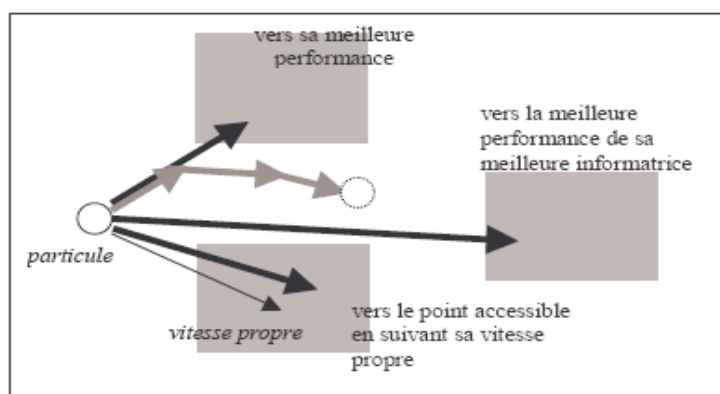


Fig. 3.5. Schéma vectorielle du déplacement d'une particule

3.3.2.7.1. Principe de base d'un PSO

Dans le PSO, chaque individu de la population est dit (particule), tandis que la population est connue sous le nom d'essaim ou «swarm». Il faut noter, que la particule peut bénéficier des mouvements des autres particules dans la même population pour ajuster sa position et sa vitesse durant le processus d'optimisation. Chaque individu utilise l'information locale à laquelle il peut accéder sur le déplacement de ses plus proches voisins pour décider

de son propre déplacement. L'espace de recherche est défini comme un « voisinage », différentes topologies sont considérées. Topologie en (étoile, rayon, circulaire, ..), le plus utilisé est le circulaire tel que schématisé sur la figure suivante (Fig.3.6) [Clerc, 2003].

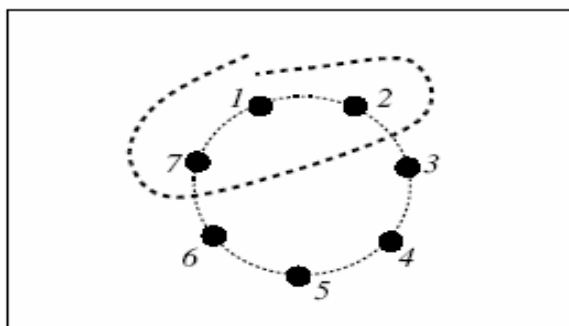


Fig. 3.6. Le cercle virtuel pour un essaim de sept particules.

Le groupe d'information de taille trois de la particule 1 est composé des particules 1, 2 et 7. Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire. Ensuite, à chaque pas de temps :

- § Chaque particule est capable d'évaluer la qualité de sa position et de garder en mémoire sa meilleure performance, c'est-à-dire la meilleure position qu'elle a atteinte jusqu'ici (qui peut en fait être parfois la position courante) et sa qualité (la valeur en cette position de la fonction à optimiser).
- § Chaque particule est capable d'interroger un certain nombre de ses congénères de son voisinage et d'obtenir de chacune d'entre elles sa propre meilleure performance.
- § A chaque pas de temps, chaque particule choisit la meilleure des meilleures performances dont elle à connaissance, modifie sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence.

A partir des quelques informations dont elle dispose, une particule doit décider de son prochain mouvement, c'est-à-dire décider de sa nouvelle vitesse. Pour ce faire, elle combine trois informations :

- § Sa vitesse actuelle.
- § Sa meilleure position actuelle.
- § La meilleure performance (vitesse et position) de ses voisines.

Le hasard joue un rôle grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l'exploration de l'espace de recherche. Naturellement, pour pouvoir

être programmé, tout ceci est formalisé dans des équations de mouvement. Un point intéressant est que, contrairement à bien d'autres heuristiques qui restent purement expérimentales, il existe une analyse mathématique précisant les conditions de convergence et le choix des paramètres [Clerc et al., 2002]

3.3.2.7.2. L'algorithme du PSO

L'algorithme de PSO commence avec une initialisation aléatoire de l'essaim de particules dans l'espace de recherche. Chaque particule est modélisée par sa position dans l'espace de recherche et par sa vitesse. A chaque instant, toutes les particules ajustent leurs positions et vitesses, donc leurs trajectoires, par rapport à leurs meilleures positions, à la particule ayant la meilleure position dans l'essaim et à leur position actuelle. En réalité, chaque particule est influencée, non seulement par sa propre expérience, mais aussi par celle des autres particules. La position et la vitesse d'une particule dans un espace de recherche à N dimensions sont définies par $P_i = (p_{i,1}, \dots, p_{i,N})$ et $V_i = (v_{i,1}, \dots, v_{i,N})$ respectivement. Chaque particule est caractérisée par sa meilleure position $L_i = (l_{i,1}, \dots, l_{i,N})$ à l'itération t . La meilleure position qu'atteint l'essaim est sauvegardée dans le vecteur $G_i = (g_1, \dots, g_N)$. La vitesse de chaque particule est mise à jour selon l'expression suivante [Clerc, et al., 2002]:

$$v_{ij}(t+1) = K[wv_{ij}(t) + c_1r_1(l_{ij} - v_{ij}(t)) + c_2r_2(g_j - v_{ij}(t))]$$

$$K = \frac{2}{|2 - j - \sqrt{j^2 - 4j}|}$$

Où $j = c_1 + c_2$ tel que $j > 4$, c_1 et c_2 sont des constantes appelées coefficients d'accélération, w est une constante appelée facteur d'inertie, r_1 et r_2 sont des nombres aléatoires uniformément distribués dans l'intervalle $[0,1]$. Si la vitesse calculée fait sortir une particule de l'espace de recherche, sa fitness n'est pas calculée. Compte tenu de la nouvelle vitesse, la position à l'itération $t+1$ est alors calculée :

$$p_{ij}(t+1) = p_{ij}(t) + v_{ij}(t+1)$$

pour $j = 1, \dots, N$

Le paramètre d'inertie w contrôle l'influence de l'ancienne vitesse sur la vitesse courante, afin de permettre aux particules d'éviter les minima locaux. De la même façon, c_1 contrôle le comportement de la particule dans sa recherche autour de sa meilleure position et c_2 contrôle

l'influence de l'essaim sur le comportement de la particule [Clerc, et al, 2002], [Guillaume C, 2009]. Les différentes étapes de l'algorithme est présenté ci-dessous (**Algorithme 3.11**).

```

1) Initialisation de la population pop "création de la population, des valeurs de seuils aléatoirement "
2) Evaluation et choix de la population pop "évaluation toutes les solutions candidates"
3) Stockage de la meilleur de la meilleur solution T* avec une fitness d'un emplacement séparé
4) mise à jour des partiales mémoires. .
    Pour un nombre d'itération fixe
      Pour tout les individus j de la population pop
        Pour toutes les solutions candidates de paramètres K de T
          "Mise à jour de la vitesse"
           $j_1 = rand(j_{min} j_{max})$  et  $j_2 = rand(j_{min} j_{max})$ 
           $pop_{vk} = w * pop_{vk} + j_1 (bestpop_{tk} - pop_{tk}) + j_2 (t_k^* - pop_{tk})$ 
          "la contrainte des vitesse est [ $v_{min} v_{max}$ ]"
          Si  $pop(j)_{vk} > v_{max}$  alors  $pop(j)_{vk} = v_{max}$ 
          Si  $pop(j)_{vk} < v_{min}$  alors  $pop(j)_{vk} = v_{min}$ 
          "mise à jour de position "
           $pop(j)_{tk} = pop(j)_{tk} + pop(j)_{vk}$ 
          "la contrainte de position de [ $g_{min} g_{max}$ ]"
          Si  $pop(j)_{vk} > g_{max}$  alors  $pop(j)_{vk} = g_{max}$ 
          Si  $pop(j)_{vk} < g_{min}$  alors  $pop(j)_{vk} = g_{min}$ 
        Fin pour " k
      Fin pour " j
      Evaluation de la population pop "évaluation tout les solutions candidates"
      " Mise à jour les partiale mémorisée"
      Pour chaque particule j
        Si la fitness est meilleur qu'avec quelques particules bestpop alors
           $bestpop(j)_{tk} = pop(j)_{tk}$ 
        Comparer les meilleures individus T de Pop avec T*
        Si T à une valeur de fitness meilleure que T* alors remplaçant T* avec T
      Fin pour " : itération
5) Production une meilleure solution enregistré T*.

```

Algorithme 3.11: Principe de l'algorithme PSO.

L'algorithme PSO à connu beaucoup de succès auprès de la communauté d'optimisation. Ses bonnes performances dans différentes applications et la possibilité d'hybridation avec d'autre métaheuristique ont contribué à cet engouement. Un état de l'art sur le PSO et ses variantes est présenté dans [Banks, et al, 2007a], [Banks, et al, 2007b]. L'algorithme PSO a été également utilisé pour résoudre le problème du seuillage [Hammouche et al., 2010]. Il affiche des meilleures performances par rapport aux autres métaheuristiques

3.3.2.8. Autres métaheuristiques

Il existe d'autres algorithmes d'optimisation métaheuristiques utilisés dans le seuillage d'histogramme tels que l'algorithme de fourrage des bactéries (bacteria foraging algorithm) [Kivin and Passino, 2002], [Maitra and Chatterjee, 2008] qui reproduit le comportement des bactéries telles que les bactéries d'E. Coli qui se trouvent dans les intestins, l'algorithme d'essaim de poissons (Artificial Fish Swarm Algorithm) [Jiang et al., 2007] qui imite le comportement des poissons et l'algorithme par luciole (Firefly algorithm) qui s'inspire sur le comportement social des lucioles (insectes ayant la propriété d'émettre de la lumière) et leur communication basé sur le phénomène de bioluminescence [Horng and Liou, 2011].

3.4. Conclusion

Dans ce chapitre, nous avons dressé un état de l'art sur les méthodes d'optimisation basées sur les métaheuristiques et qui ont été utilisées dans le seuillage. Ces méthodes sont divisées en deux classes: les méthodes à solution unique et les méthodes à population de solutions. Nous avons ensuite choisi et détaillé la méthode d'essaims de particules (PSO) pour sa précision, sa simplicité et sa vitesse de convergence dans le but de résoudre le problème de seuillage d'histogrammes multidimensionnels.

Chapitre 4

Tests et résultats

4.1 Introduction

Après avoir présenté un état de l'art sur les méthodes de segmentation par seuillage d'histogramme et un état de l'art sur quelques méthodes d'optimisation métaheuristique dans les chapitres précédents, nous exposerons dans ce chapitre les résultats des quelques méthodes de seuillage implémentées.

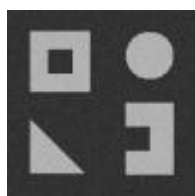
Afin de mieux évaluer les nombreuses méthodes programmées et de valider les résultats obtenus, nous avons choisi plusieurs images tests. Ces images sont des images en niveaux de gris de taille 256x256 codées sur 8bits.

4.2. Critères d'évaluation

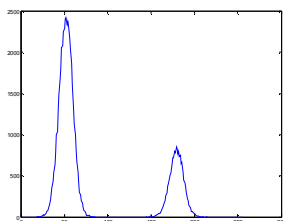
La comparaison est basée sur le calcul du temps et de l'évaluation de la qualité de la segmentation en utilisant un critère d'évaluation. Ce critère d'évaluation est classé en deux catégories [S. Philipp et al., 2006] en fonction de la disponibilité ou non d'une référence image segmentée. Dans la première catégorie, une évaluation supervisée est considérée, où l'utilisateur ou un expert fournit une image de référence segmentée. Un critère appartenant à cette catégorie est utilisé dans le présent document: le taux de classification T_c qui exprime le pourcentage de pixels correctement classifiés. La segmentation est la meilleure quand T_c est proche de 100%.

4.3. Images synthétiques

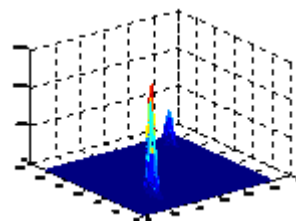
Il s'agit dans cette première application d'évaluer les résultats des différentes méthodes de seuillage. Pour cela nous avons utilisé des images synthétiques ayant respectivement 2, 3 et 4 classes et dont les histogrammes monodimensionnels et bidimensionnels sont donnés sur la figure (4.1).



(a1)



(b1)



(c1)

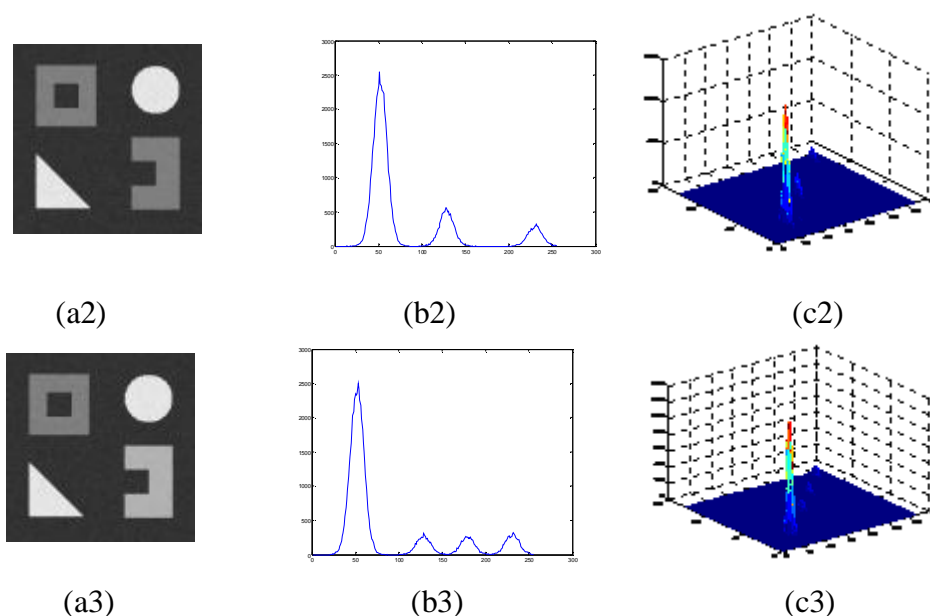


Fig. 4.1 : (a) : Image synthétique 1, (b) : Image synthétique 2, (c) Image synthétique 3, 1 : image. 2 : histogrammes monodimensionnels, 3 : histogrammes bidimensionnels.

4.3.1. Calcul exhaustif des seuils

Nous présenterons dans cette section les résultats de la recherche exhaustive des seuils à partir des histogrammes monodimensionnels et bidimensionnels. Notons que dans le cas d'un histogramme bidimensionnel ou tridimensionnel, le calcul des seuils par une recherche exhaustive prend un temps prohibitif et qu'il nous a été impossible de le relever surtout lorsque le nombre de seuils est élevé.

Les tableaux (4.1) à (4.3) récapitulent les valeurs des seuils, des fitness et deux critères d'évaluations obtenus par les méthodes entropiques et par la méthode d'Otsu. Ces tableaux regroupent également le temps de calcul des seuils consommé par chaque méthode.

Méthodes	Histogrammes	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	mono	(65)	10.2250	4.104337	96.5775
	bi	(85, 113)	17.4930	48.241645	99.9863
	tri	(88, 112, 80)	15.7929	333.901860	99.9619
Otsu	mono	(85)	3.0728e+003	0.002972	100
	bi	(80, 89)	5.9480e+003	42.700226	98.2971
	tri	(88, 92, 72)	9.0661e+003	1043.510431	98.1857
Renyi	mono	(67)	10.7156	0.265099	98.0392
	bi	(85, 113)	18.8081	217.989156	99.9863
	tri	(88, 116, 80)	17.8222	1151.588423	99.9710
Shannon	mono	(65)	7.0875	0.011948	96.5775
	bi	(85, 113)	12.1252	55.301519	98.0392
	tri	(88, 112, 80)	10.9855	955.548022	99.9619
Tsallis	mono	(67)	80.0162	0.013198	98.0392
	bi	(85, 113)	1.3530e+003	222.785441	99.9908
	tri	(88, 116, 80)	8.9794 e+003	1045.121370	99.9863

Tableau 4.1: Résultats du seuillage de l'image synthétique 1 d'histogrammes basés sur la recherche exhaustive

Méthodes	Histogrammes	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	mono	(66),(140)	15.4296	4.114337	97.2592
Otsu	mono	(85),(162)	3.0054e+003	0.441269	99.9893
Renyi	mono	(67),(142)	16.1148	3.795846	97.8836
Shannon	mono	(66),(140)	10.6950	2.345183	97.2592
Tsallis	mono	(69),(144)	5.2459e+003	4.732727	98.8449

Tableau 4.2: Résultats du seuillage de l'image synthétique2 d'histogrammes basés sur la recherche exhaustive

Méthodes	Histogramme	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	mono	(68),(140),(207)	20.3951	20.557818	98.4924
Otsu	mono	(87),(153),(205)	3.5911e+003	94.456643	99.9878
Renyi	mono	(68),(141),(198)	21.3017	232.905653	98.5107
Shannon	mono	(68),(140),(207)	14.1368	19.563498	98.4924
Tsallis	mono	(68),(141),(198)	6.5054e+003	5.681198	98.5107

Tableau 4.3: Résultats du seuillage de l'image synthétique 3 d'histogrammes basés sur la recherche exhaustive

Les figures (4.2) et (4.4) montrent quelques images segmentées obtenus par le seuillage d'histogrammes monodimensionnels et bidimensionnels et tridimensionnels.



(a1)



(a2)



(a3)



(b1)



(b2)



(b3)



(c1)



(c2)



(c3)

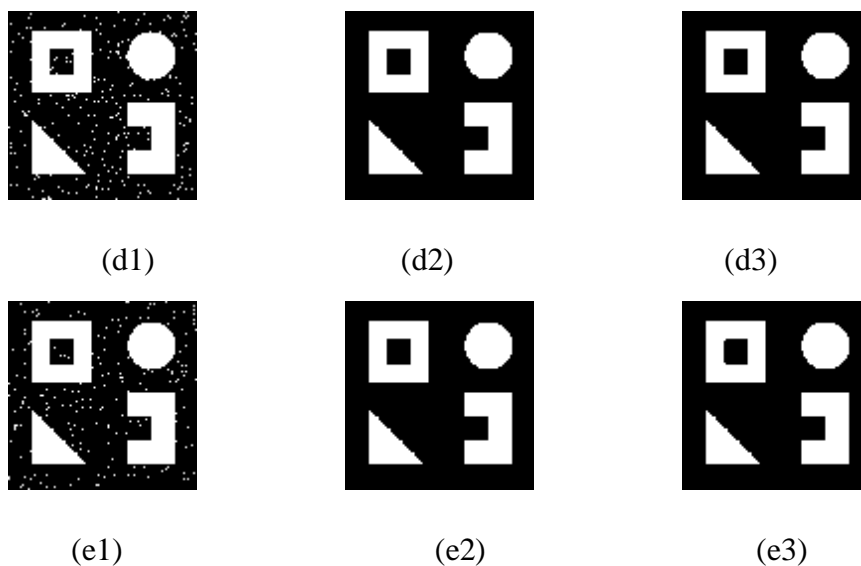
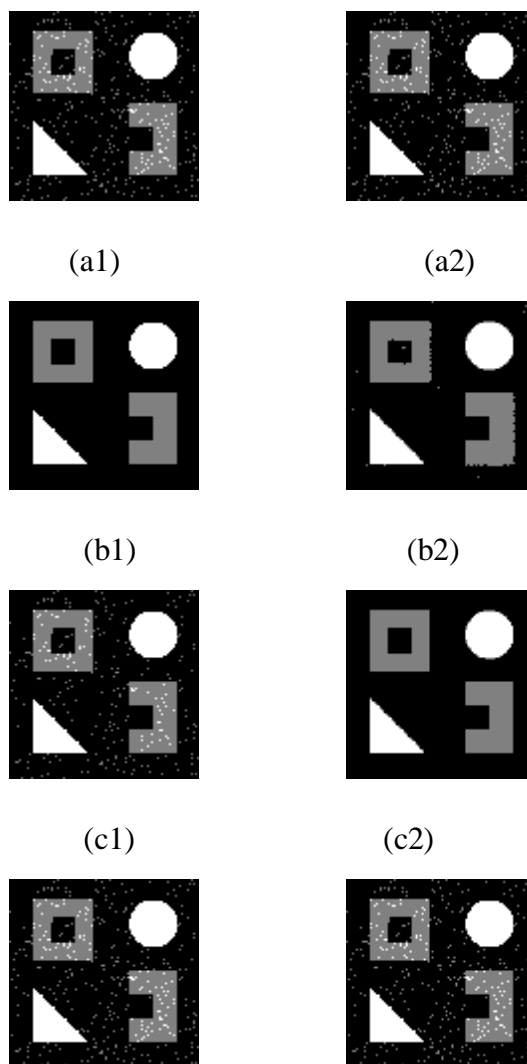


Fig. 4.2 : Image synthétique 1 segmentée par seuillage d’histogrammes par la recherche exhaustive.

(a) : Kapur, (b) : Otsu (c) Renyi (d) Shannon (e) Tsallis. Colonne 1 : histogramme monodimensionnel, Colonne 2 : histogramme bidimensionnel, Colonne 3 : histogramme tridimensionnel.



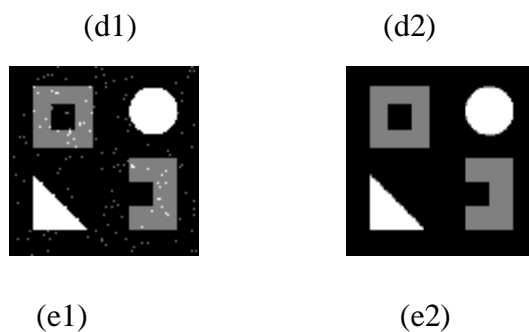


Fig. 4.3 : Image synthétique 2 segmentée par seuillage d'histogrammes par la recherche exhaustive. (a) : Kapur, (b) : Otsu (c) Renyi (d) Shannon (e) Tsallis. Colonne 1 : histogramme monodimensionnel, Colonne 2 : histogramme bidimensionnel.

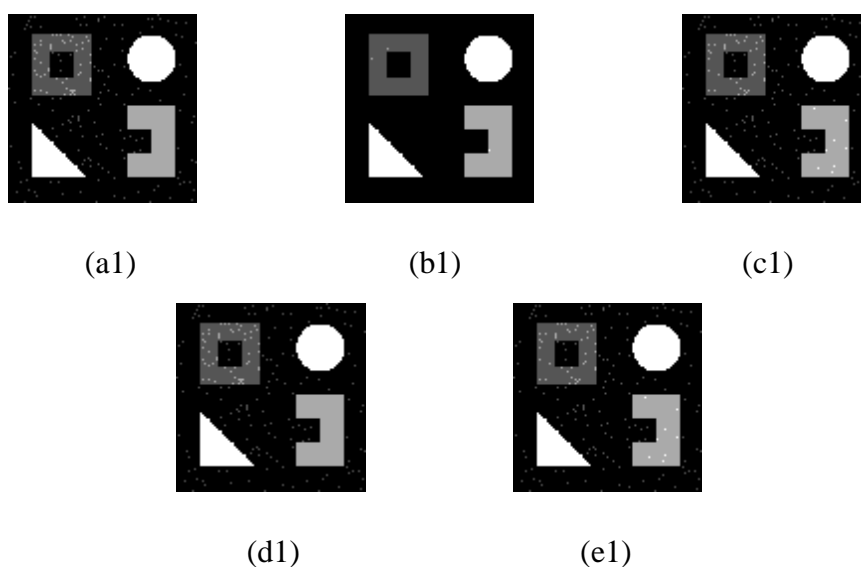


Fig. 4.4: Image synthétique 3 segmentée par seuillage d'histogrammes par la recherche exhaustive. (a) : Kapur, (b) : Otsu (c) Renyi (d) Shannon (e) Tsallis. Colonne 1 : histogramme monodimensionnel.

4.3.2. Interprétation des résultats

Le tableau 4.1 montre bien que le temps de calcul augmente lorsqu'on passe d'un histogramme monodimensionnel à un histogramme bidimensionnel puis à un histogramme tridimensionnel.

Dans le cas de l'image synthétique 1, le tableau 4.1 montre également que hormis la méthode d'Otsu, les résultats de la segmentation basés sur l'histogramme bidimensionnel sont proches de ceux obtenus par l'histogramme tridimensionnel est meilleurs que ceux obtenus par l'histogramme monodimensionnel.

La méthode d'Otsu suivie des méthodes de Renyi et Tsallis donnent de meilleurs résultats lorsque l'histogramme monodimensionnel est utilisé. Dans le cas de l'histogramme bidimensionnel et tridimensionnel, les résultats fournis par toutes les méthodes sont proches avec un léger avantage pour la méthode de Tsallis.

4.3.3. Calcul des seuils par la méthode d'essai de particules

Pour rechercher rapidement les seuils à partir des histogrammes mono ou multidimensionnels, nous avons implémenté l'algorithme d'essai de particules (PSO) présenté dans le chapitre précédent.

4.3.3.1. Paramètres du PSO

Les paramètres de l'algorithme PSO dépendent principalement du nombre de seuils et de type d'histogramme (mono, bi et tridimensionnel).

Nous avons déterminé expérimentalement les valeurs de ces paramètres. Ces paramètres sont les mêmes pour chaque fonction objective et pour un même type d'histogramme (tableau. 4.4).

Histogrammes	Fonction objective				
	Kapur	Otsu	Renyi	Shannon	Tsallis
Monodimensionnels	N_pop=1000 maxit=100 c1=1 c2=2-c1 C=1	N_pop=1000 maxit=100 c1=1 c2=2-c1 C=1	alpha=0.5 N_pop=1000 maxit=100 c1 =1 c2=2-c1 C=1	N_pop=1000 maxit=100 c1 =1 c2=4-c1 C=0.1	alpha=0.5 N_pop=1000 maxit=100 c1 =1 c2 =2-c1 C=1
Bidimensionnels	N_pop=700 maxit=100 c1 =1 c2=4-c1 C=0.1	N_pop=1000 maxit=100 c1 =1 c2=2-c1 C=1	alpha=0.5 N_pop=700 maxit=100 c1 =1 c2=2-c1 C=1	N_pop=700 maxit=100 c1 =1 c2=2-c1 C=1	alpha=0.5 N_pop=700 maxit=100 c1 =1 c2=2-c1 C=1
Tridimensionnels	N_pop=500 maxit=100 c1 =1; c2=2-c1 C=1	N_pop=500 maxit=100 c1 =1 c2=2-c1 C=1	alpha=0.5 N_pop=500 maxit=100 c1 =1 c2=2-c1 C=1	N_pop=500 maxit=100 c1 =1 c2=4-c1 C=0.1	alpha=0.5 N_pop=500 maxit=100 c1 =1 c2=2-c1 C=1

Tableau. 4.4: Paramètres de l'algorithme PSO

4.3.3.2. Résultats du PSO

Les tableaux (4.5) à (4.7) donnent les résultats du seuillage obtenus par l'algorithme PSO à partir des différents histogrammes et par chaque fonction objective.

Méthodes	Histogrammes	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	Mono	(65)	10.2167	4.104337	96.5775
	Bi	(106, 113)	17.4930	265.579749	99.9863
	Tri	(118,116,147)	22.5651	12725.243358	99.9725
Otsu	Mono	(85)	3.0731e+003	4.357899	100
	Bi	(80, 89)	5.9480e+003	236.675118	98.2971
	Tri	(134, 94, 102)	8.9849e+003	6808.895396	98.2147
Renyi	Mono	(67)	10.7187	5.365286	98.0392
	Bi	(104,113)	18.8081	446.076151	99.9863

	Tri	(136,116, 134)	23.6748	18370.41638	99.9725
Shannon	Mono	(65)	7.0878	6.279506	96.5775
	Bi	(91,93)	12.6913	557.779661	98.9410
	Tri	(128,116, 81)	15.6409	4040.928068	99.9710
Tsallis	Mono	(67)	80.1018	5.069792	98.0392
	Bi	(90,95)	1.7166e+003	867.101321	99.4019
	Tri	(136,114, 134)	7.3167e+003	17370.412638	99.9680

Tableau 4.5: Résultats du seuillage d’image synthétique1 par l’algorithme PSO

Méthodes	Histogrammes	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	Mono	(66),(140)	15.4296	18.053163	97.2595
	Bi	(95,93), (195,141)	26.1637	330.680791	99.4385
	Tri	(85,93,76), (194,140,164)	33.2447	5314.115749	99.2264
Otsu	Mono	(92),(171)	3.0054e+003	4.796692	99.9893
	Bi	(78,79), (182,141)	5.8294e+003	291.216396	99.1379
	Tri	(85,79,91), (153,144,172)	8.8055e+003	7160.913156	97.6868
Renyi	Mono	(67),(142)	16.1148	11.440374	97.8836
	Bi	(71,74), (192,139)	27.8134	484.860447	97.7478
	Tri	(95,93,101), (166,141,185)	34.5445	2457.9746	99.2264
Shannon	Mono	(66),(140)	10.6950	18.655123	97.2595
	Bi	(88,92), (166,144)	16.1519	110.067688	99.4263
	Tri	(88,93,84), (189,140,173)	23.0435	2642.388106	99.2264
Tsallis	Mono	(67),(142)	673.6891	12.239290	97.8836
	Bi	(86,94), (186,140)	2.1315e+004	213.243731	99.4446
	Tri	(103,93,101), (205,136,148)	5.9163e+005	369.545130	99.2264

Tableau 4.6: Résultats du seuillage d’image synthétique2 par l’algorithme PSO

Méthodes	Histogrammes	Seuils	Fitness	Temps (s)	Taux de classification
Kapur	Mono	(68),(140),(207)	20.3951	20.557818	98.4924
	Bi	(87,96),(154,134), (205,187)	34.2252	174.461170	99.3408
	Tri	(97,96,89), (154,134,144), (205,187,213)	43.3291	3698.910751	98.7839
Otsu	Mono	(98),(153),(205)	3.5911e+003	5.568196	99.9878
	Bi	(80,87),(151,134),(204,187)	6.8068e+003	133.485644	98.7976
	Tri	(78,94,68), (158,135,156), (210,207,210)	1.0168e+004	6235.044895	98.6679
Renyi	Mono	(68),(141),(198)	21.3017	12.073489	98.5107
	Bi	(99,96),(154,134), (205,187)	36.1400	251.740721	99.3408
	Tri	(99,96,81), (154,160,149), (205,187,196)	44.6155	1853.645285	98.7839
Shannon	Mono	(68),(140),(207)	14.1368	19.563498	98.4924
	Bi	(87,96),(154,134), (205,187)	23.7231	208.851879	99.3408
	Tri	(96,96,102), (152,134,140), (205,187,208)	30.0332	2378.591913	98.7839
Tsallis	Mono	(68),(141),(198)	6.5054e+003	5.681198	98.5107
	Bi	(98,96),(154,134), (205,187)	1.8411e+006	254.448545	99.3408
	Tri	(91,96,95), (154,134,141), (205,187,198)	4.0235e+007	3946.567055	98.7839

Tableau 4.7: Résultats du seuillage d’image synthétique3 par l’algorithme PSO

Les figures (4.5) à (4.7) montrent quelques images segmentées par seuillage d’histogrammes mono, bi et tridimensionnels

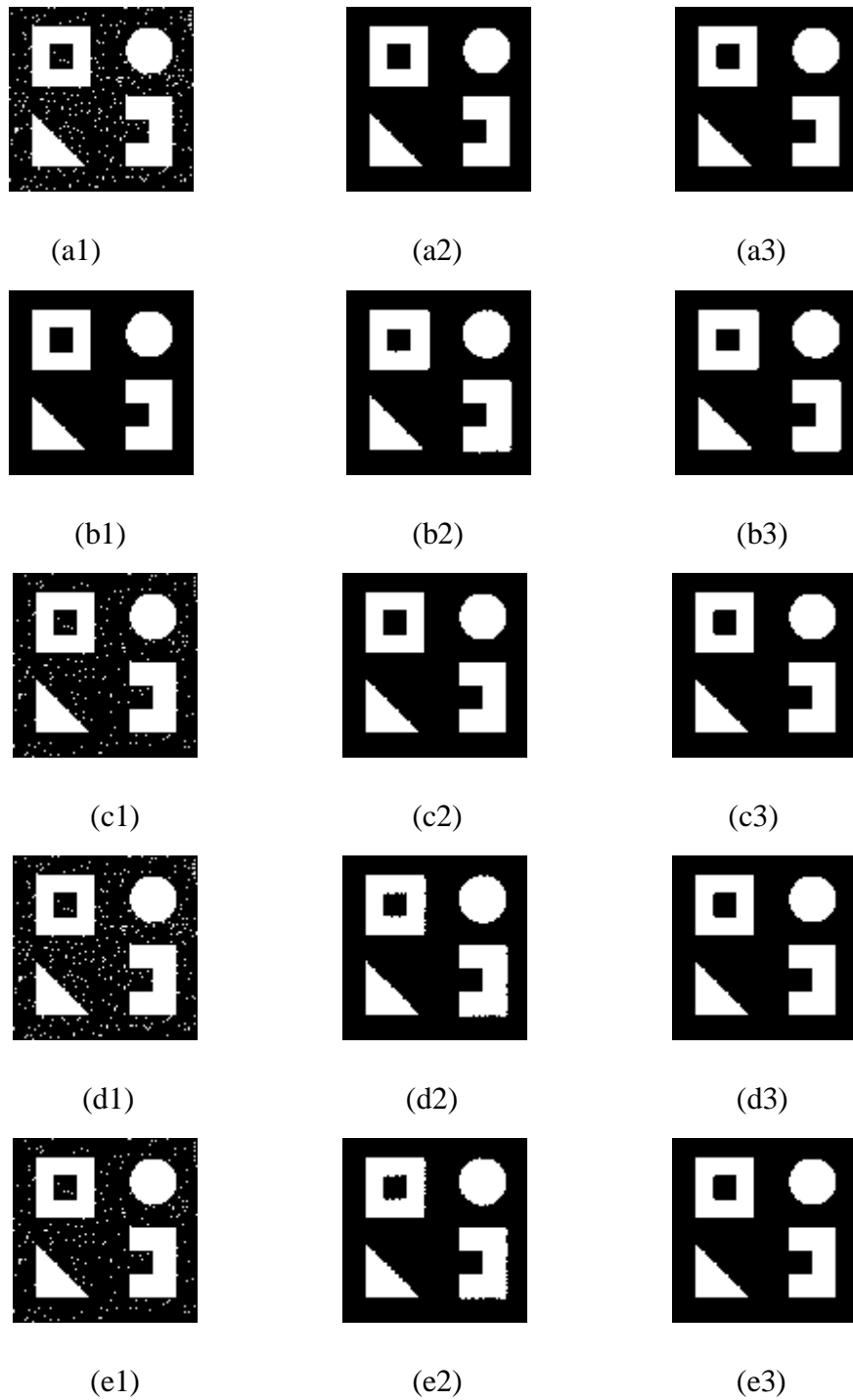
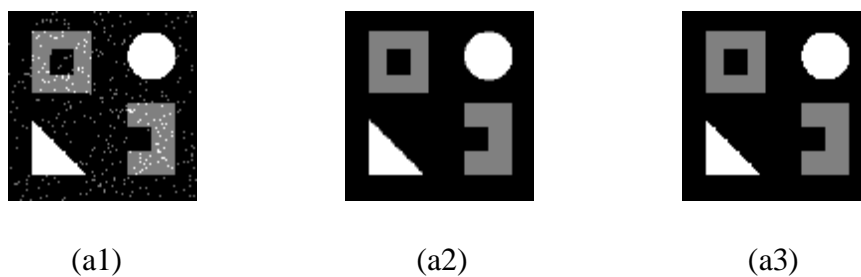


Fig. 4.5 : Image synthétique 1 segmentée par seuillage d'histogrammes basé sur le PSO. Kapur, (b) : Otsu, (c) : Renyi, (d):Shannon, (e): Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel



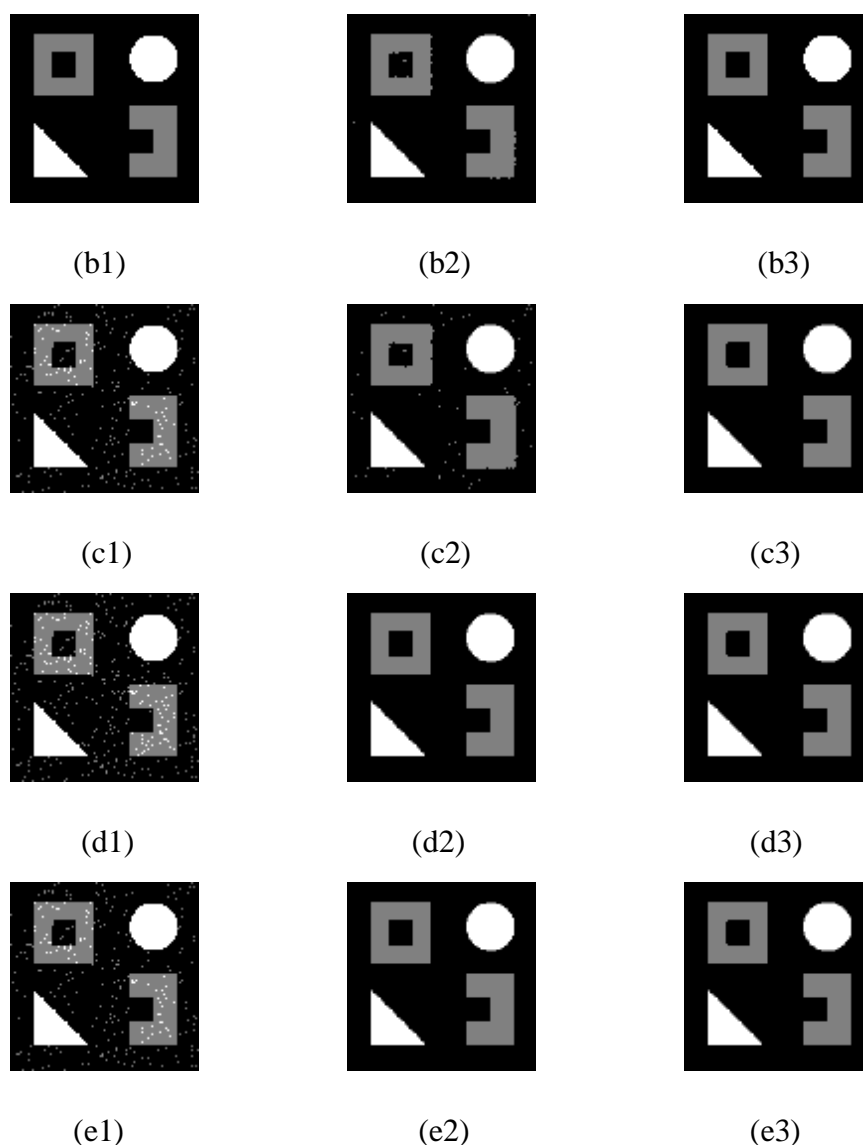
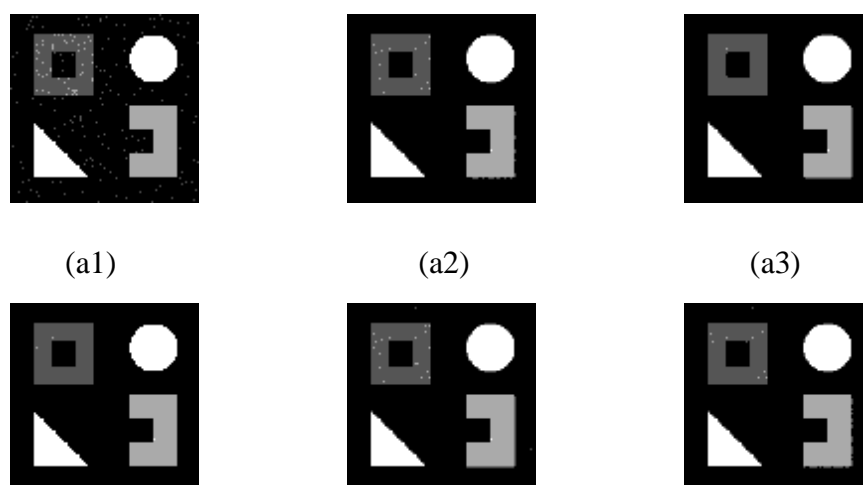


Fig. 4.6 : Image synthétique2 segmentée par seuillage d’histogrammes basé sur le PSO. Kapur, (b) : Otsu, (c) : Renyi, (d):Shannon, (e): Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel



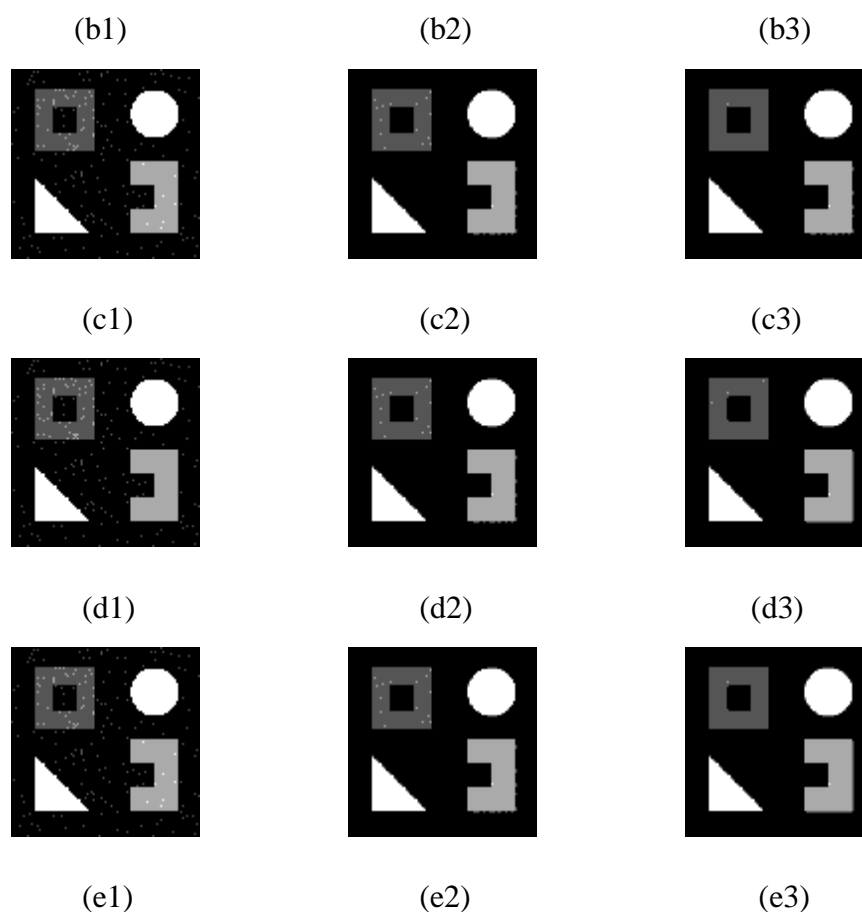


Fig. 4.7 : Image synthétique3 segmentée par seuillage d’histogrammes basé sur le PSO. Kapur, (b) : Otsu, (c) : Renyi, (d):Shannon, (e): Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel

4.3.3.3. Interprétation des résultats

Les tableaux précédents montrent que les valeurs des seuils de segmentation des trois images synthétiques obtenus par l’algorithme PSO sont proches de ceux obtenus par la recherche exhaustive, ce qui confirme la convergence du PSO. Les tableaux montrent également que, grâce à l’algorithme PSO, le temps de calcul des seuils sur un histogramme monodimensionnel et bidimensionnel est moins élevé par rapport a celui obtenu par la recherche exhaustive et le taux de classification est plus élevé dans le cas des histogrammes bi et tridimensionnels par rapport aux histogrammes monodimensionnels. Le PSO, contrairement à la recherche exhaustive, nous permet dans le cas des histogrammes multidimensionnels de déterminer les seuils même lorsque leur nombre est élevé.

Lorsqu’on cherche un seul seuil en utilisant un histogramme monodimensionnel, la méthode d’Otsu apparait comme la plus performante alors que c’est les méthodes de Kapur, Reny et un degré moins Tsallis qui donnent un meilleur résultat lorsqu’on cherche un couple de seuils en utilisant l’histogramme bidimensionnel. En utilisant l’histogramme

tridimensionnel, toutes les méthodes donnent presque le même taux de classification qui est légèrement supérieur à celui obtenu par la méthode d'Otsu.

Lorsque le nombre de seuil est égal à 2 et dans le cas d'un histogramme monodimensionnel, la méthode d'Otsu qui donne le meilleur taux de classification. En utilisant un histogramme bidimensionnel, les résultats du tableau (4.7) montrent que la méthode de Tsallis apparait plus performante alors que dans le cas d'un histogramme tridimensionnel, toutes les méthodes donnent le même taux de classification qui est supérieur à celui obtenu par la méthode d'Otsu.

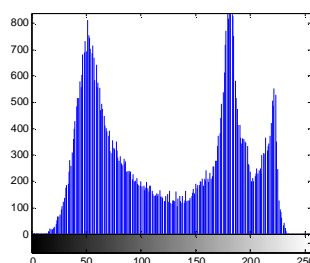
Pour l'image synthétique ou le nombre de seuils est égal à 3, la méthode d'Otsu donne de meilleurs résultats lorsqu'on utilise un histogramme monodimensionnel. Pour un histogramme bidimensionnel et tridimensionnel, toutes les méthodes ont donné le même taux de classification qui est supérieur à celui obtenu par la méthode d'Otsu.

4.4. Images réelles

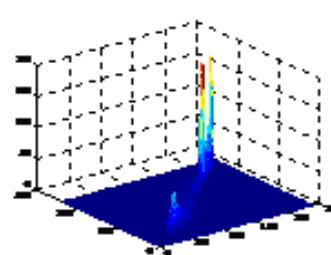
Nous allons maintenant présenter les résultats de la segmentation par seuillage d'histogrammes obtenues sur d'autres images réelles de tailles 256X256 codées sur 8 bits. La Figure (4.9) montre ces images ainsi que leurs histogrammes mono et bidimensionnel.



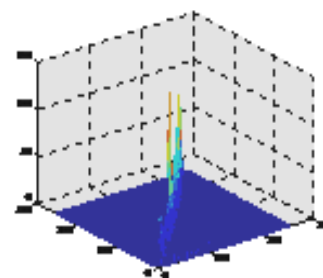
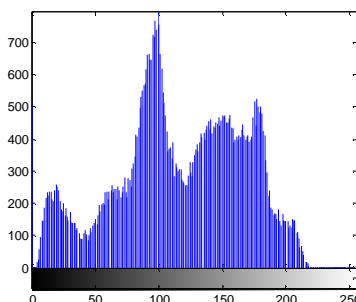
(a1)



(a2)



(a3)



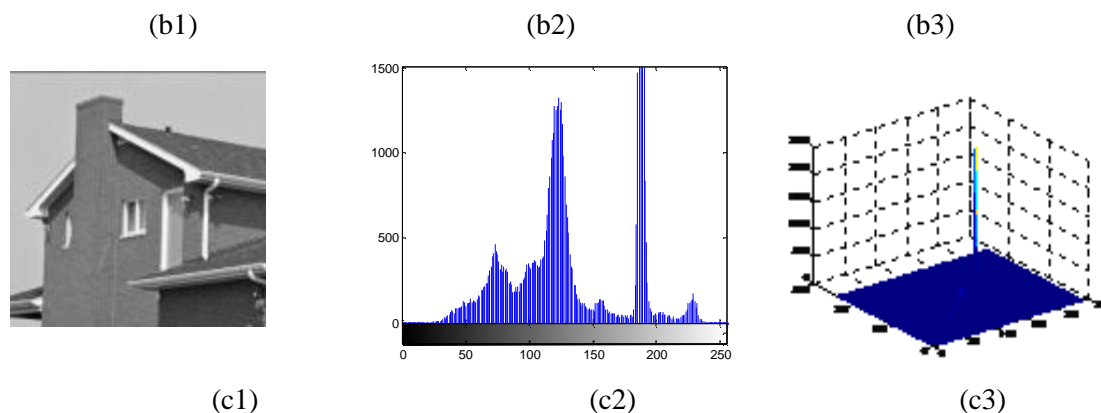


Fig. 4.8 : Images tests. (a) : Lac, (b): Pepper (c) House (1) : image, (2) : Histogrammes monodimensionnels, (3) : Histogrammes bidimensionnels.

4.4.1. Paramètres du PSO pour les images réelles

Les paramètres de l'algorithme PSO dépendent principalement du nombre de seuils et de type d'histogramme (mono, bi et tridimensionnel) de chaque image.

Nous avons déterminé expérimentalement les valeurs de ces paramètres. Ces paramètres sont les mêmes pour chaque fonction objective et pour un même type d'histogramme de chaque image. Sauf pour la taille de la population qui varie selon le nombre de seuils et selon la dimension de l'histogramme (tableau. 4.8).

Histogrammes	Nombre de seuils	Fonction objective				
		Kapur	Otsu	Renyi	Shannon	Tsallis
Monodimensionnels	2	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1
	4	N_pop=2000 c2=2-c1 C=1	N_pop=2000 c2=2-c1 C=1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1
	5	N_pop=2000 c2=2-c1 C=1	N_pop=3000 c2=2-c1 C=1	N_pop=1000 c2=2-c1 C=1	N_pop=1000 c2=2-c1 C=1	N_pop=1000 c2=2-c1 C=1
Bidimensionnels	2	N_pop=1000 c2=4-c1 C=0.1	N_pop=2000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1
	4	N_pop=3000 c2=2-c1 C=1	N_pop=5000 c2=2-c1 C=1	N_pop=1000 c2=2-c1 C=1	N_pop=2000 c2=2-c1 C=1	N_pop=3000 c2=2-c1 C=1
	5	N_pop=3000 c2=2-c1 C=1	N_pop=6000 c2=2-c1 C=1	N_pop=4000 c2=2-c1 C=1	N_pop=3000 c2=2-c1 C=1	N_pop=4000 c2=2-c1 C=1
Tridimensionnels	2	N_pop=2000 c2=4-c1 C=0.1	N_pop=2000 c2=4-c1 C=0.1	N_pop=2000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1	N_pop=1000 c2=4-c1 C=0.1
	4	N_pop=6000 c2=2-c1 C=1	N_pop=2000 c2=2-c1 C=1	N_pop=4000 c2=2-c1 C=1	N_pop=4000 c2=2-c1 C=1	N_pop=4000 c2=2-c1 C=1
	5	N_pop=8000 c2=2-c1 C=1	N_pop=3000 c2=2-c1 C=1	N_pop=5000 c2=2-c1 C=1	N_pop=5000 c2=2-c1 C=1	N_pop=6000 c2=2-c1 C=1

Tableau. 4.8: Paramètres de l'algorithme PSO

4.4.2. Résultats du PSO

Les tableaux (4.9) à (4.11) donnent les résultats du seuillage obtenus par l'algorithme PSO à partir des différents histogrammes de chaque fonction objective et de chaque image. Chaque image est segmentée avec un nombre de seuils fixé au préalable et dont le choix a été fait d'une manière subjective après avoir examiné chaque image.

Méthodes	Histogrammes	Seuils	Temps (s)	Fitness
Kapur	Mono	(92), (163)	14.45721	17.9856
	Bi	(87,92), (157,149)	407.159950	31.9380
	Tri	(73,79,75), (146,138,144)	1745.753748	37.3344
Otsu	Mono	(88), (155)	3.483261	3.7617e+003
	Bi	(92,147) (201,200)	215.692886	7.0607e+003
	Tri	(115,130,92) (159,155,159)	2534.734282	1.0647e+004
Renyi	Mono	(92),(163)	8.447215	18.2653
	Bi	(86,92), (158,148)	40.118424	30.5501
	Tri	(92,96,92) (160,144,152)	1267.991768	32.8069
Shannon	Mono	(92) (163)	11.350498	12.4667
	Bi	(87,92), (157,149)	280.680730	22.1377
	Tri	(73,79,75) (146,138,144)	3479.316369	21.1222
Tsallis	Mono	(92), (163)	8.329222	1.5671e+003
	Bi	(85,90),(154,146)	386.516195	3.6050e+005
	Tri	(73,78,73) (146,138,144)	4742.31519	2.3624e+006

Tableau 4.9 : Résultats du seuillage d'image Lac par l'algorithme PSO

Méthodes	Histogrammes	Seuils	Temps (s)	Fitness
Kapur	Mono	(58), (104), (145), (186)	12.901522	26.5897
	Bi	(55,57), (95,102), (139,136), (186,177)	300.509442	45.9652
	Tri	(58), (104), (145), (186)	12.901522	26.5897
Otsu	Mono	(44), (83) (121),(162)	6.579424	2.4472e+003
	Bi	(73,73)(105,118)(131,130)(160,159)	193.180253	4.3276e+003
	Tri	(75,76,67), (132,123,111) (196,184,195), (228,216,230)	1094.65976	5.9071e+003
Renyi	Mono	(49), (90), (138), (185)	7.095835	26.9445
	Bi	(42,48)(84,88) (136,130)(186,172)	243.453597	43.8090
	Tri	(48,56,52), (88,100,96) (140,132,136) (188,172,180)	1378.137389	45.2334
Shannon	Mono	(48), (85), (137), (186)	12.670742	18.4321
	Bi	(48,50), (82,86), (131,130), (186,174)	223.097073	31.8575
	Tri	(59,59,59) (103,98,100) (142,139,141) (185,175,179)	1339.466455	37.6395
Tsallis	Mono	(49), (90), (138), (185)	5.115872	7.8489e+004
	Bi	(47,50) (84,89) (133,130) (184,173)	288.476231	2.3445e+008
	Tri	(63,64,64), (107,107,107), (142,140,142), (184,177,179)	1391.25732	4.8681e+009

Tableau 4.10 : Résultats du seuillage d'image Peppers par l'algorithme PSO

Méthodes	Histogrammes	Seuils	Temps (s)	Fitness
Kapur	Mono	(62), (102), (138), (183), (195)	15.286469	28.6178
	Bi	(52, 56) (90,94)(116,130) (160,152)(196,182)	87.996571	47.4260
	Tri	(43,48,45), (85,884,92), (118,117,127), (159,155,156), (194,183,190)	1068.164821	59.0669
Otsu	Mono	(80), (109), (137), (170), (206)	7.027743	2.0807e+003
	Bi	(90,92), (111,113) (147,131), (162,149) (204,197)	308.041996	3.8520e+003
	Tri	(50,45,48), (91,87,89) (115,113,110) (148,166,157) (192,198,192)	510.241926	5.3053e+003
Renyi	Mono	(60),(99),(137) (177) ,(197)	1.205320	29.4734
	Bi	(60,62), (91,93)(117,129) (159,153) , (198,183)	243.116351	54.4850
	Tri	(43,44,39), (73,77,78), (113,116,127),	1280.774204	60.6885

		(165,162,175), (205,184,206)		
Shannon	Mono	(62)(101) (138), (182) (195)	1.752692	19.8272
	Bi	(54,59)(87,92) (120,129)(161,157) (196,183)	226.018014	54.3352
	Tri	(49,50,47), (85,80,82), (116,120,119), (157,151,153), (190,183,201)	1287.82143	39.8204
Tsallis	Mono	(60),(99), (137), (173), (198)	7.104988	2.5118e+005
	Bi	(59,62)(95,98) (129,130) (160,160)(197,183)	220.909215	3.7713e+009
	Tri	(42,44,43), (81,83,77), (114,113,114), (147,146,148), (199,180,199)	1189.02853	4.3301e+010

Tableau 4.11: Résultats du seuillage d’image House par l’algorithme PSO

Les figures (4.9) à (4.11) montrent quelques images segmentées par seuillage d’histogrammes mono, bi et tridimensionnels



(a1)



(a2)



(a3)



(b1)



(b2)



(b3)



(c1)



(c2)



(c3)



(d1)



(d2)



(d3)



Fig. 4.9 : Image Lac segmentée par seuillage d'histogrammes basé sur le PSO. (a) : Kapur, (b) : Otsu, (c) : Renyi, (d):Shannon, (e): Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel





(e1)

(e2)

(e3)

Fig. 4.10 : Image peppers segmentée par seuillage d’histogrammes basé sur le PSO Kapur, (b) : Otsu, (c) : Renyi, (d):Shannon, (e): Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel



(a1)



(a2)



(a3)



(b1)



(b2)



(b3)



(c1)



(c2)



(c3)



(d1)



(d2)



(d3)

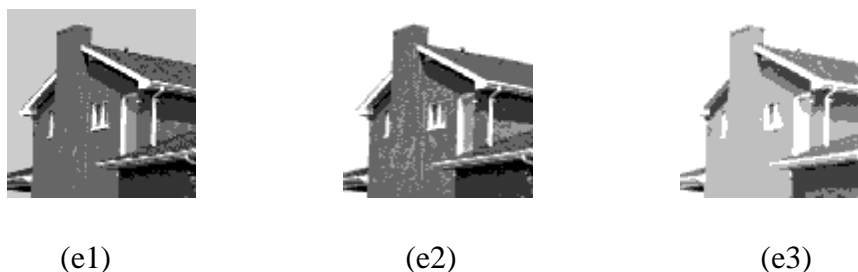


Fig. 4.11 : image segmentées par seuillage d’histogrammes basé sur le PSO. (a) : Kapur, (b) : Otsu (c) : Renyi (d) : Shannon (e) : Tsallis. (1) : monodimensionnel, (2) : bidimensionnel, (3) : tridimensionnel

4.4.3. Interprétation des résultats

En consultant visuellement les images segmentées des figures 4.9 à 4.11, on peut remarquer que la méthode d’Otsu apparaît comme la plus performante dans le cas des trois images tests en utilisant l’histogramme monodimensionnel et que dans le cas de l’histogramme bidimensionnel toutes les méthodes donnent presque les mêmes images segmentées, différentes de celles obtenues par la méthode d’Otsu. Par contre les méthodes de Kapur, Reny et Shannon donnent des meilleurs résultats lorsqu’on utilisant l’histogramme tridimensionnel.

4.5. Conclusion

Nous avons évalué dans ce chapitre quelques méthodes de seuillage d’histogrammes monodimensionnel, bidimensionnel et tridimensionnel. Pour accélérer de recherche d’un ou plusieurs seuils, la méthode d’essaim de particules a été utilisée. Elle permet de donner presque les mêmes résultats que la recherche exhaustive mais avec un temps de calcul raisonnable. Les résultats de cette méthode ont été illustrés sur des exemples d’images synthétiques et réelles. Ils montrent l’intérêt d’utiliser des histogrammes multidimensionnels. Ils montrent également qu’une méthode de seuillage peut être performante dans le cas du seuillage d’un histogramme de dimension donnée et moins performante dans le cas d’un histogramme ayant une autre dimension. La méthode d’Otsu par exemple est la plus performante lorsqu’elle est appliquée avec un histogramme monodimensionnel et ce quelque soit le nombre de seuils recherchés. Par contre, elle devient moins performante dans le cas des histogrammes bi et tridimensionnels.

Conclusion générale

Conclusion générale

Dans ce mémoire nous avons présenté une panoplie de méthodes de segmentation basées sur le seuillage global d'histogrammes. Ces méthodes appartiennent à l'approche non paramétrique qui consiste à déterminer les seuils en optimisant une fonction objective.

Nous avons ainsi rappelé certaines techniques de seuillage qui sont basées soit sur un histogramme monodimensionnel bidimensionnel et tridimensionnel de l'image soit sur l'entropie de cet histogramme.

Cependant l'utilisation de l'histogramme monodimensionnel peut s'avérer dans certaines situations (images bruitées, mauvaise contraste) inadaptée. Afin de remédier à cet inconvénient, nous nous sommes intéressés au seuillage d'histogrammes bi et tridimensionnels. Ces techniques utilisées pour le seuillage d'histogrammes monodimensionnels ont été alors étendues au cas multidimensionnels.

Cependant il existe des images pour lesquelles il est impossible d'effectuer une bonne segmentation au moyen d'un seuil unique, cette difficulté est liée aux problèmes de l'inhomogénéité de la réponse du capteur (camera), aux effets d'ombre et à la nature des images à traiter. Afin d'y remédier, les méthodes de seuillage d'histogrammes bidimensionnels et tridimensionnels ont été généralisées.

Le problème des méthodes de seuillage bidimensionnel et tridimensionnel et du multiseuillage est le temps d'excursions. Pour accélérer les calculs nous avons implémenté une méthode d'optimisation métaheuristique à savoir l'algorithme d'essaim de particule (PSO).

Les tests menés sur divers image montrent l'intérêt d'utiliser les histogrammes multidimensionnels.

Ce travail peut être enrichi en exploitant les mêmes techniques employées dans le seuillage sur d'autres types d'histogrammes bidimensionnels ou sur des histogrammes tridimensionnels dans le cadre de la segmentation d'image couleur

Comme, il est possible d'améliorer la rapidité de la segmentation par seuillage d'histogrammes bidimensionnels et tridimensionnels en appliquant des versions améliorées du PSO ou d'autres métaheuristicques.

Bibliographie

Bibliographie

- [Abbasss, 2001] H.A. Abbasss, Marriage in honey bee optimization (HBO): a haplometrosis polygynous swarming approach, in: The Congress on Evolutionary Computation (CEC001), pp. 207–214, 2001.
- [Abutaleb, 1989] A. S. Abutaleb. "Automatic Thresholding of gray-level pictures using two dimensional entropy. Computer Vision, Graphics and Image Processing." 1989, vol. 47, pp. 22-32. 1989.
- [Azerradj et Kessoum, 2007] D. Azerradj et D. Kessoum. "Application des contours actifs géodésiques pour la segmentation de la vidéo ". PFE. INI. 2006/2007. Mémoire de fin d'études, INI 2007.
- [Azarbad et al., 2011] M. Azarbad, A. Ebrahimzade, V. Izadian Segmentation of Infrared Images and Objectives Detection Using Maximum Entropy Method Based on the Bee Algorithm International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM). ISSN: Vol. 3, pp. 026-033, 2011.
- [Bachelet, 1999] V. Bachelet. Métaheuristiques parallèles hybrides : Application au QAP. PhD thesis, USTL LIFL France, 1999.
- [Bäck, 1993] T. Bäck, H-P. Schwefel, An overview of evolutionary algorithms for parameter optimization. Evolutionay Computation vol.1 n°1, pp 1-24, 1993.
- [Banks, et al., 2007a] A. Banks, J. Vincent, and C. Anyakoha, A review of particle swarm optimization. Part I: background and development. Nat. comput., pp. 1-18. 2007.
- [Banks, et al., 2007b] A. Banks, J. Vincent, and C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat. comput., pp. 1-16. 2007.
- [Boucher, 1998] A. Boucher, C. Garbay, "Des agents spécialisés pour la compréhension de séquences d'images ", RFIA. 98, vol.II, pp.275-284, 1998.
- [Bernsen, 1986] J. Bernsen, Dynamic thresholding of gray level images, ICPR'86: Proc. Intl. Conf. Patt. Recog., pp. 1251–1255, 1986.
- [Beveridge et al., 1989] J.R. Beveridge, J.G. Riffith. R. R. Kohler A.R. Hanson et E. M. Riseman Segmenting images using localized histograms and region merging IJC. vol.2, pp. 311-347, 1989.
- [Campbell, 1996] L.L. Campbell, Exponential entropy as a measure of extent of a distribution Probability Th. and Related Fields, vol. 5, n° 3, pp. 217-225, 1996.
- [Canny, 1986] J. Canny, A computational approach to edge detection, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 8, n°6, pp. 679-698, 1986.
- [Castan et al., 1989] S.Castan, J. Zhao and J. Shen. Une famille de détecteurs de contours basée sur filtre exponentiel.optimal. In AFCET-RFIA, 1989.

- [Chang et al., 1995] F.J.Chang, J.C.Yen and S.Chang, A new criterion for automatic multilevel thresholding, IEEE trans. Image process. vol 4. pp. 370-378, 1995.
- [Chen et al., 1999] S. Chen, and B. L. Luk, Adaptive simulated annealing for optimization in signal processing applications. Signal Processing., vol.79, pp. 117-128, 1999.
- [Chung et al., 2005] F. Wang Shitong; L. Chung. Note on the equivalence relationship between Renyi-entropy based Tsallis-entropy based image thresholding. Pattern Recognition Letters. pp. 2309-2312, 2005.
- [Chehdi, 1991. 1] K. Chehdi and D. Coquin, Binarisation of various images by Detecting local Threshold with A Validation Test. IEEE Pacific RIM Conference on Communications Computers and Signal Processing, pp. 611-614, 1991.
- [Chehdi, 1991. 2] K. Chehdi and D. Coquin, Binarisation d'images par seuillage local optimal maximisant un critère d'homogénéité, Troisième colloque GRETSI-Juan-les-pins, pp. 1096-1072, 1991.
- [Chilali, 2006] O. Chilali. Classification automatique de données utilisant les modèle deformable. Thèse de magister, université mouloud Mammeri, Tizi Ouzou, Algerie, 2006.
- [Chang, 1994] C.I. Chang, K. Chen, J. Wang and M.L.G. Althouse, a relative entropy based approach to image thresholding, Pattern recognition vol.27.n° 9, pp.1275-1289, 1994.
- [Clerc, 2003] M. Clerc, TRIBES, Un exemple d'optimisation par essaim particulaire sans paramètre de contrôle, Conférence OEP'03, Paris, France, 2003.
- [Clerc, et al., 2002] M. Clerc, and J. Kennedy, The particle swarm, explosion, stability, and convergence in multi-dimensional complex space. IEEE Trans. on Evolutionary Compt, vol.6, pp. 58-73. 2002.
- [Cocquerez et al., 1995] J.P Cocquerez and S. Philipp. Analyse d'images : filtrage et segmentation. Masson, Paris, 1995.
- [Creutz, 1983] Creutz, M., Microcanonical Monte Carlo simulation. Physical Review Letters., vol.50, n°19, pp. 1411-1414. 1983.
- [Cuevas et al., 2010] Erik cuevas, Daniel Zaldivar, Marco Pérez-Cisneros, A novel multi-threshold segmentation approach based on differential evolution optimization, Expert systems with application vol.37, pp. 5265-5271, 2010.
- [Cuevas et al., 2009] Erik cuevas, Daniel Zaldivar, Marco Valentine Osuna-enciso, Marco Pérez-Cisneros. A novel multi-threshold segmentation approach based on Artificial Immune System Optimization Advances in computational Intell., AISC 61, n°1, pp. 309-37. 2009.
- [Deravi et al., 1983] F.Deravi and S.K.Pal, Gray level thresholding using second-order statistics, Pattern recognition letters, pp. 417-422, 1983.
- [De Castro and Von Zuben, 2000] L. De Castro, and F. Von Zuben, Artificial Immune Systems. A Survey of Applications. Technical Report DCA-RT 02/00, Department of Computer Engineering

and Industrial Automation, Part II, School of Electrical and Computer Engineering, State University of Campinas, Brazil. 2000.

[Dempster, et al., 1977] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*. vol. 39, n°1, pp.1-38, 1977.

[Deriche, 1987] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, *International Journal of Computer Vision*, pp. 167-187, 1987.

[Deriche, 1990] R. Deriche. Fast algorithms for low level vision. *IEEE Trans. Pattern Analysis and Machine Intelligence* vol.12 n°1. pp. 78-87.1990.

[Diaf et al., 2009] M. Diaf, K. Hammouche, P. Siarry. From the real ant to the artificial ant: application in combinatorial optimization, data clustering, collective robotics and image processing. *Nature-Inspired Informatics for Intelligent: Applications and Knowledge Discovery: Applications in Business, Science and Engineering*, IGI Eds. USA, pp. 298-322, July 2009.

[Digabel et al., 1978] H. Digabel et C. Lantuéjoul. Iterative algorithms Dans J. L. Chermant, éditeur, *Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, Stuttgart, Riederer Verlag. pp.85-99, 1978.

[Dréo et al., 2003] J. Dréo, A. Pérowski, P. Siarry, E. Taillard, *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003.

[Dorigo, et al., 1996] M. Dorigo, V. Maniezzo, and A. colorni, Ant System optimization by a colony of cooperating agents. *IEEE Trans. on Man. Cyber. Part B.*, vol. 26, n°1, pp. 29-41. 1996.

[Eusuff et Lansey, 2003] M.M. Eusuff, K.E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm. *J. Water Resour. Planning Manag.* vol. 129 n°3, pp. 210–225. 2003.

[Glover, 1986] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, vol.13, pp. 533–549, 1986.

[Gonzalez et Woods, 1992] R.C. Gonzalez And R. E. Woods. *Digital image processing*. Addison-Wesley, Reading, MA. 1992.

[Gonzalez et Woods, 2002] R. C. Gonzalez, and R. E. Woods, *Digital image processing*. Pearson Prentice Hall, 2002.

[Gong et al., 1998] Jina Gong, Liyuan Li and Weinan Chen Fast Recursive Algorithms for Two-Dimensional thresholding, PII: S0031-3203 (97) 00043-5, *Pattern Reconition*, vol.31, n°3, pp. 295-300, 1998.

[Calas, 2009] Guillaume Calas, *Optimisation par essaim de particules [SCIA]* EPITA, rue Voltaire, 94270 Le Kremlin-Bicêtre, France. pp.14-16 2009.

- [Haralick, 1973] R. M. Haralick, K. Shanmugam and I. Dinstein, "texture feature for image classification", IEEE Trans. Syst., Man and Cybernetics, SMC, vol 3, pp. 610-621, 1973.
- [Hammouche et al., 2010] K. Hammouche, M. Diaf, P. Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, journal home page 2009.
- [Hammouche et al., 2008]: Hammouche, K., Diaf, M., Siarry, P., A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Computer Vision Image Understanding vol.109, n°2, pp.163–175. 2008.
- [Hansen, 1986] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy. 1986.
- [Havrda, Charvat, 1967] J. Havrda, F. Charvat, Quantification methods of classification processes: Concept of structural entropy. Kybernetika (Prague) vol.3, pp. 95–100. 1967.
- [Herault et al., 1993] L. Herault, and R. Horaud, Figure Ground Discrimination, A combinatorial optimization approach. IEEE Trans. on Pattern Ana. and Mach. Intelli., vol.15, n°9, pp. 899 - 914. 1993.
- [Horwitz and Pavlidis, 1976] S.L. Horowitz, and T. Pavlidis. Picture segmentation by a tree traversal algorithm. Journal of the association for computing machinery, vol.23 n°2, pp. 368-388, 1976.
- [Holland, 1975] J. Holland Adaptation in Natural and Artificial Systems, University of Michigan Press MIT, 1975.
- [Horng, 2010] M. H. Horng. Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. Expert Systems with applications. vol. 37, pp. 4580-4592, 2010.
- [Horng, 2009] M. H. Horng, , Ting-Wei Jiang and Jin-Yi Chen Member, IAENG Optimization, Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009, IMECS Hong Kong, vol. I pp.18-20, 2009.
- [Horng and Liou, 2011] M. Horng et R. Liou, Multilevel minimum Cross entropy threshold selection based on the firefly algorithm. Expert system with application, vol. 38, pp 14805-14811, 2011.
- [Jain et al., 1999] A. K. Jain, M. N. Murty and P. J. Flynn. Data clustering: a review. ACM Computing Surveys, vol. 31, n°3, pp.264, 322, 1999.
- [Jiang et al., 2007] M. Jiang, N. Mastorakis, D. Yuan, M. A. Lagunas, Multi-threshold Image Segmentation with Improved Artificial Fish Swarm Algorithm Block-coding and Antenna Selection, European Computing Conference (ECC), 2007
- [Jing et al., 2003] X. Jing, J. Li, Y. Liu, Image segmentation based on 3-D maximum between cluster variance. Acta Electron. Sin. vol. 31, n°9, pp.1281–1285. 2003.

- [Jong, 1993] K.A. De Jong, W. Spears, On the state of evolutionary computation. Proc. Of International Conference on Genetic Algorithms (ICGA'93), pp. 618-623, 1993.
- [Kapur et al., 1985] J. N. Kapur, P. K. Sahoo, and A. C. K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics and Image Processing, vol. 29, pp. 273 - 285. 1985.
- [Kass et al., 1988] M. Kass A Wikin, D, Terzopoulos " Snakes: Active contour models" computer vision, graphics and image processing: pp.321-331,1988.
- [Kennedy, et al., 1995] J. Kennedy, and R. C. Eberhart, Particle swarm optimization. IEEE Int. Conf. on Neuronal networks, Piscataway (Japan). vol. 4, pp. 1942- 1948, 1995.
- [Kirsch, 1971] R. Kirsch. Computer determination of the constituent structures of biomedical images. Computer and Biomedical Research. USA, vol. 4, n° 3, pp 315-328, 1971.
- [Kittler et al., 1986] J. Kittler, and J. Illingworth, Minimum error thresholding. Pattern Recognition, vol. 19, n°1, pp. 38-52. 1986.
- [Kirkpatrick et al., 1983] S. Kirkpatrick, D.C. Gelatt, and M.P.Vechhi. Optimization by simulated annealing. Science, vol. 220, pp. 671–680, 1983.
- [Kivin and Passino, 2002] Kevin. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine, vol. 6, pp. 52-67, 2002.
- [Kohonen, 1984] T. Kohonen Self-Organization and Associative Memory Springer-Verlag, 1984
- [Laurence, 1999] G. Laurence. Trois principes de coopération pour la segmentation en imagerie de résonance magnétique cérébrale. PhD Thesis, Imag, Université Joseph Fourier Grenoble I. TIMC, 1999.
- [Leandro et al., 2002] Leandro N. De Castro and Jonathan Timmis. Artificial Immune Systems A New Computational Intelligence Approach. Springer-Verlag, 2002.
- [Li et al., 1993] C. H. LI and C. K. Lee, Minimum cross entropy thresholding. Pattern recognition.vol 26 n°4, pp. 617-625, 1993.
- [Liane, 1997] C. Liane, Ramac and K. Pramod, Varshney Image Thersholding based on Ali-Silvey distance Measures, Pattern Recognition, vol. 30, n°7, pp. 1161-1174, 1997.
- [Liu, et al., 1993] J.Z. Liu and N.Q. Li, The automatic thresholding of grey-level pictures via two-dimensional Otsu method, Acta automat. Sinica (in Chinese) .vol 19, pp.101-105, 1993.
- [Ma et al., 2008] M. Ma, Y. Zhang, H. Tian. And Y. Lu. A Fast SAR image segmentation algorithm based on particle swarm optimization and grey entropy. Fourth International Conference on Natural Computation. DOI 10.1109/ICNC.2008.577 , IEEE, 2008.
- [Mardia and Hainsworth, 1988] K.V. Mardia and T.J. Hainsworth, A special thresholding method for image segmentation, IEEE trans. Pattern analysis and Mach. Intellegence, vol.10, n°6, pp.919-927, 1988.

- [Maitra and Chatterjee, 2008] M. Maitra, A. Chatterjee. A novel technique for multilevel optimal magnetic resonance brain image thresholding using bacterial foraging, *Measurement* vol. 41 pp. 1124–1134, 2008.
- [Najman and Couprie, 2003] L. Najman, and M. Couprie, Watershed algorithms and contrast preservation. *Discrete geometry for computer imagery.*, vol. 2886, pp. 62-71. 2003.
- [Na Wang et al., 2010] Na Wang, Xia Li, Xiao-hong Chen Fast three-dimensional Otsu thresholding with shuffled frog-leaping algorithm, *Pattern Recognition Letters*, vol. 31 pp. 1809–1815, 2010.
- [Nakib 2007. 1] Nakib, A. Conception de métaheuristiques d’optimisation pour la segmentation d’images. Application à des images biomédicales thèse doctorat, 2007.
- [Nakib et al., 2007. 2] A. Nakib, H. Oulhadj, and P. Siarry, Microscopic image segmentation with two dimensional exponential entropy based on hybrid microcanonical annealing. *IAPR Conference on Machine Vision Applications (MVA)*. Tokyo (Japan), pp. 420-423, 2007.
- [Nakib et al., 2007. 3] A. Nakib, S. Roman, H. Oulhadj, and P. Siarry, Fast MRI segmentation based on two dimensional survival exponential entropy and particle swarm optimization. In *proc. of the 29th Annual Int. Conf. on the IEEE. EMBS*. Lyon, pp.5563-5565, 2007.
- [Nakib et al., 2007. 4] H. Oulhadj, A. Nakib et P. Siarry. Segmentation d’images par maximisation de l’entropie à deux dimensions basée sur le recuit microcanonique, *Université de Paris XII, Laboratoire Images, Signaux et Systèmes Intelligents, (LISSI, E. A. 3956), 61 avenue du Général De Gaulle, 94010 Créteil, France*, 2007.
- [Niblack, 1986] W. Niblack, *An Introduction to Image Processing*, Prentice-Hall, Englewood Cliffs, pp. 115–116, NJ, 1986.
- [Otsu, 1979] N. Otsu, A threshold selection method for grey level histograms. *IEEE Transactions on System, Man and Cybernetics SMC* vol. 9, n°1, pp. 62–66, 1979.
- [Ou et Lin, 2006] Chao Ou et Weixing Lin, Comparison between PSO and GA for Parameters Optimization of PID Controller, China, *Proceedings of the IEEE*. 2006.
- [Prasanna et al., 2004] P.K. Prasanna K. Sahoo, Gurdial Arora, thresholding method based on two-dimensional Renyi’s entropy *Pattern Recognition*, vol. 37, pp. 1149 – 1161, 2004.
- [Pavlidis et Liow, 1990] Pavlidis and Y-T. Liow. “integrating region growing and edge detection“ *IEEE Trans. Pattern analysis and machine intelligence*. vol. 12, n°3, pp. 225-233, 1990.
- [Portes et al., 2004] M. portes de Albuquerque I.A esquef , A.R. Gesualdi mello M. portes de Albuquerque; image thresholding using tsallis entropy *pattern recognition letters* vol. 2, n°5, pp. 1059 □1065, 2004.
- [Prewitt, 1970] J.M.S. Prewitt. Object enhancement and extraction. In *PPP*, vol. 70, pp 75– 149, 1970.

- [Price and Storn, 1997], K. Price, R. Storn, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces Report TR95 -012, International Computer Science Institute, Berkeley, California. Journal of Global Optimization vol. 11, pp. 341–359, 1997.
- [Ray et al., 1997] Ray, John R. and Harris, Richard W., Simulated annealing in the microcanonical ensemble. Physical Review E., vol. 55, n°5, pp. 5270 - 5274, 1997.
- [Roberts, 1965] L.G. Roberts. Machine perception of 3-d solids. In OE-OIP, vol. 65, pp.159–197, 1965.
- [Rosenfeld and Kak, 1981] A. Rosenfeld and A.C. Kak, Digital Picture Processing, Edition Academic Press, INC. 1981.
- [S. Philipp et al., 2006] S. Philipp-Foliquet, L. Guigues, Evaluation of image segmentation: state of the art, new criteria and comparison, Traitement du Signal Vol. 23 n°2.pp. 109-124. 2006.
- [Sahoo et al., 2004] P. K. Prasanna, K. Sahoo, and G. Arora, A thresholding method based on two dimensional Renyi's entropy. Pattern Recognition, vol. 37, pp. 1149 - 1161. 2004.
- [Sahoo et al., 2006] K. Sahoo, P. K. Prasanna and G. Arora, Image thresholding using two dimensional Tsallis-Havrda - Charvat entropy. Pattern Recognition Letters, vol. 27, pp. 520 - 528, 2006.
- [Sahoo et al., 1997] P. K. Prasanna, K. Sahoo, Carrye Wilkins and Jerry Yeager, threshold selection using renyi,s entropy pattern recognition vol. 30. n°1 PP. 71 □ 84, 1997.
- [Schoenauer et al., 1997] M. Schoenauer, Z. Michalewicz, Evolutionary computation: an Control and Cybernetics, vol. 26, n°3, pp. 307-338, 1997.
- [Sobel, 1978] I. Sobel. Neighbourhood coding of binary images for fast contour following and general array binary processing. Computer Graphics and Image Processing, vol.8, pp. 127-135, 1978.
- [Shen et al., 1986] J. Shen and S. Castan. An optimal linear operator for edge detection. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'86), Miami Beach, Florida, pp. 109--114, USA, 1986.
- [Shen et al., 1992] J. Shen and S. Castan_ An optimal linear operator for step edge detection. CVGIP. Graphics Models and Image Processing. 54(2). 112-133, Mars 1992.
- [Synder, et al., 1990] W. Synder, G. Bilbro, A. Logenthiran, and S. Rajala, Optimal thresholding A new approach. Pattern Recognition Letters. vol. 11, pp. 803 - 810, 1990.
- [Strasters and Gerbrands, 1991] K. C. Strasters and J. J. Gerbrands, Three-dimensional image segmentation using a split, merge and group approach," Pattern Recognition Letters, vol. 12, pp. 307-325, 1991.
- [Talbi et al., 2009] E.G. Talbi and A. Liefoghe, L. Jourdan. A unified model for evolutionary multi-objective optimization and its implementation in a general purpose software framework. In

IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (IEEE MCDM), , Nashville, Tennessee, USA. IEEE Press, pp. 88-95, 2009.

[Thiria et al., 1997] S. Thiria, Y. Lechevallier, O. Gascuel and S. Canu, *Statistique et Méthodes Neuronales*, Dunod, Paris, 1997.

[Tao et al., 2003] W.B. Tao, Jin-Wen Tian, Jian Liu Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm *Pattern Recognition Letters*, vol. 24 pp.3069–3078. 2003.

[Ohlander et al., 1979] Ohlander, K. Price, and D. R. reddy “picture segmentation using recursive region splitting method” *computer graphics and image processing*, vol, 8, pp. 313-333, 1979.

[Venkateswarlu, 1995] N.B.Venkateswarlu, Implimentation of some image thresholding algorithms on a connection machine-200, *Pattern recognition letters*. vol 16, pp .759-768, 1995.

[Weska, et al., 1974] J.S. Weska, R.N. Nagel and A. Rosenfeld, A threshold selection technic. *IEEE trans. On computers*, vol.23, pp.1322-1326, 1974.

[Woods et al., 1998] I. Woods, and T. Downs, Demon algorithms and their application to optimization problems. In *Proc. of Int. Joint Conf. on Neural Networks 2 and IEEE World Congress on Computational Intelli.*, Anchorage (USA). pp. 1661 - 1666, 1998,

[Wang et al., 2009] Lei Wang, Shen Ting-zhi Two-Dimensional Entropy method based on genitic algorithm, Dept. of Electronic Engineering, Beijing Institute of Technology, Beijing, 100081, China, 2009.

[Yang et al., 2003] Z.-H. Yang, Z.-B. Pu, Z.-Q. Qi, Relative entropy multilevel thresholding method based on genetic algorithm, in: *IEEE International Conference on Neural Networks and Signal Processing*, Nanjing, China, pp. 583–586, 2003.

[Yin end Chen, 1997] P. Yin, and Y. Chen, L.-H.A fast iterative scheme for multi level thresholding methods. *Signal Processing Vol. 60*, pp. 305–313, 1997.

[Yu and Fan, 2008] Hai-Yan Yu and Jiu-Lun Fan Three-Level image segmentation based on maximum fuzzy partition entropy of 2-D histogram and quantum genetic algorithm, *Département of information and control Xi’an institute of post and telecommunication*, LNAI 5227, pp 484-403, 2008

[Zucker, 1976] S. Zucker, Region growing Childhood and adolescence. *Computer Graphics and Image Dans Processing* 5, pp. 382-399, 1976.

[Z. Hong Qing, 2004] Z. HongQing. segmentation of blood vessels in retinal image using 2-D entropies of gray-level gradient co-occurrence matrix. *ICASSP*, 2004.

[Zhang and Zhang, 2006] Y.F. Zhang and Y. Zhang. Another method of building 2D entropy to realize automatic segmentation. *Journal of Physics: Conference Series*, vol. 48, pp. 303–307, 2006.