

*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département Automatique*



Mémoire

De fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état

En Automatique

THEME

Initialisation automatique des snakes

Pour une segmentation d'une image IRM.

Dirigé par :

M^{elle} : CHILALI O.

Présenté par :

M: AKLI KAMAL

M: HAMID SAMIR

Devant le jury:

M: Hammouche K,

M^{me}: Hammouche S.

M: Hasnaoui.

Promotion 2007-2008

Remerciements

En premier lieu, nous tenons à remercier notre promotrice, M^{elle} CHILLALI pour avoir accepté de nous encadrer, pour sa patience et sa disponibilité qui nous a énormément aidé pour que ce travail puisse voir le jour.

Nous remercions sincèrement les membres du jury pour avoir accepté de porter un avis sur ce travail.

Enfin, nous remercions nos familles pour leur soutien, leurs encouragements et leurs présences tous au long de nos cursus universitaires.

Merci à tous.

Dédicaces

Je dédie ce travail :

A la mémoire de ma mère que Dieu l'accueille en son saint paradis ;

A mon cher père qui m'a offert son aide ;

A mes très chers frères ;

A mes très chères sœurs ;

A ma meilleure amie et compagne Djedjiga ;

A mon ami frère Hamid et sa femme ;

A tous mes amis ;

A tous mes collègues d'études ;

A tous ceux qui me sont cher(e)s ;

Samir.

Dédicaces

Je dédie ce travail :

A la meilleure maman au monde pour son soutien et son amour ;

A mon cher père qui m'a offert tout son aide ;

A mes très chers frères ;

A mes très chères sœurs ;

A tous mes amis : Momoh, Faiz, Said et Hamid ;

A tous mes collègues d'études ;

A tous ceux qui me sont cher(e)s.

Kamal.

SOMMAIRE

Introduction générale	1
------------------------------------	---

CHAPITRE I

Généralités

I.1 Introduction	4
I.2 Caractéristiques d'une image numérique	4
I.2.1 Pixel	4
I.2.2 Dimension	5
I.2.3 Résolution	5
I.2.4 Luminance	5
I.2.5 Contraste	5
I.2.6 Contour	5
I.2.7 Texture	6
I.2.8 Bruit	6
I.3 Histogramme	6
I.3.1 Définition	6
I.3.2 Types d'histogrammes	7
I.3.3 Les différentes opérations appliquées sur l'histogramme	7
<i>a) Modification d'histogramme</i>	7
<i>b) Egalisation d'histogramme</i>	8
<i>c) Etirement d'histogramme</i>	8
<i>d) Inversion d'images (négatif)</i>	9
I.4 Le seuillage	9
I-4-1 Le seuillage fondé sur l'analyse de l'histogramme	10
I.4.2 Le seuillage adaptatif (local)	10
I.4.3 La technique globale	10
I.5 Filtrage	10
I.5.1 Définition	10
I.5.2 Filtrage linéaire	11
I.5.3 Filtrage adaptatif ou non linéaire	11
I.5.4 Filtrage morphologique	11
I.6 La segmentation	14
I.6.1 Définition	14
I.6.2 Approches de la segmentation	15
I.6.2.1 Approche par classification des pixels	15

SOMMAIRE

I.6.2.2 Approche régions	15
I.6.2.3 Approche frontières	15
I.7 Détection des contours	15
I.7.1 Les approches dérivatives	16
I.7.1.1 Approche gradient	16
a) Calcul direct	17
b) Opérateur de Roberts	17
c) Opérateur de Sobel	18
d) Opérateur de Prewitt	19
e) Masque dérivé de gaussienne	19
I.7.1.2 Les filtres optimaux	20
I.7.1.3 Approche Laplacien	20
I.7.2 Les méthodes surfaciques	22
I.7.3 Les contours actifs	22
I.8 Conclusion	22

CHAPITRE II

Contours actifs

II.1 Introduction	24
II.2 Principe des contours actifs	24
II.3 Formulation mathématique	25
II.3.1 Energie interne	25
II.3.2 Energie externe	26
II.3.3 Processus de minimisation	26
II.3.3.1 Méthodes d'optimisation locales	27
II.3.3.2 Méthodes d'optimisation globale	28
II.4 Implémentation des <i>snakes</i> par méthode variationnelle	29
II.5 Exemple	33
II.6 Intérêts et limites des <i>snakes</i>	34
II.7 La force de ballon	35
II.8 Flux du vecteur gradient	36
II.8.1 Calcul de la carte de contours	37
II.8.2 Le calcul du champ GVF	37
II-9 Flux du vecteur gradient généralisé	39

SOMMAIRE

II.9.1 Principe et définition.....	39
II.9.2 Choix des fonctions de pondération.....	39
II.9.3 Calcul du champ GGVF.....	40
II.10 contours actifs géométriques et géodésiques.....	41
II.11 conclusion.....	42

CHAPITRE III

Approche adoptée

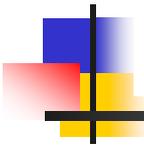
III.1 Introduction.....	44
III.2 Langage de programmation.....	44
III.3 Approche adoptée.....	44
III.4 Implémentation.....	45
III.4.1 Lecture de l'image et Calcul de la carte contour.....	46
III.4.2 interpolation et reparamétrisation.....	46
III.4.3 Choix du <i>snake</i> classique, GVF ou GGVF.....	47
III.4.4 Calcul de la carte contour.....	47
III.4.5 Calcul du champ GVF.....	47
III.4.6 Calcul du champ GGVF.....	48
III.4.7 Déformation du <i>snake</i>	49
III.5 Critère d'arrêt.....	49
III.6 Choix des paramètres de l'approche.....	50
a) Terme d'élasticité.....	50
b) Terme de rigidité.....	50
c) Le pas du temps.....	50
d) Le poids du champ GVF.....	50
III.7 Initialisation automatique.....	51
III.7.1 Analyse de l'histogramme.....	51
III.7.2 Sélection de la classe de la région d'intérêt.....	52
III.7.3 Détection de contour.....	53
III.7.4 Sélection de la région d'intérêt.....	54
III.8 Conclusion.....	55

SOMMAIRE

CHAPITRE IV

Tests et résultats

IV.1 Introduction	58
IV.2 Application de l'approche sur l'image test	58
IV.2.1 Initialisation manuelle	58
IV.2.2 Initialisation automatique	60
IV.2.3 Interprétation des résultats	61
IV.3 Application de l'approche sur une image IRM	62
IV.3.1 La tumeur	63
IV.3.2 Détection de tumeurs cérébrales	63
IV.3.3 Initialisation manuelle	65
IV.3.4 Initialisation automatique	66
IV.3.5 interprétation des résultats	67
IV.4 conclusion	69
Conclusion générale	71
Annexe	
Bibliographie	



Introduction générale

Introduction générale

Le progrès des moyens informatiques et des techniques de traitement ont permis d'ouvrir une voie de développement très prometteuse vers le traitement d'images.

Les techniques du traitement d'images ont été développés, avec succès, dans divers domaines, car l'objectif attendu de ce traitement est l'extraction d'informations pertinentes, sur les objets présents (leurs formes, leurs nature, dimensions, vitesses, etc.). Il s'agit, donc, d'un domaine très vaste qui trouve de plus en plus d'applications, notamment, dans la reconnaissance d'objets, l'imagerie satellitaire, l'imagerie médicale, etc.

Cependant, malgré ces progrès significatifs, l'image reste aujourd'hui, encore, un sujet difficile et les solutions actuelles ne permettent pas, toujours, de résoudre, de manière satisfaisante, de nombreux problèmes importants. Il est souvent évident, pour n'importe lequel d'entre nous, de dessiner les bords d'un objet dans une image. En revanche, dans toute sa généralité, cette tâche de segmentation est très difficile à réaliser, automatiquement, par des moyens informatiques. C'est un paradoxe bien connu du traitement de l'image où les tâches, que l'homme réalise naturellement, sont souvent les plus difficiles à automatiser. Au delà de cette remarque, la segmentation constitue une étape fondamentale dans le traitement et l'analyse d'images. On dénombreait déjà plusieurs approches de segmentation.

Récemment, de nouvelles approches, regroupées sous le nom de modèles déformables, ont été proposées dans la littérature. Ces approches se limitent, généralement, sur des propriétés comme la continuité ou la régularité, qui caractérise le contour d'un objet. Dans ce cas, on parlera plutôt de contours actif, ou *snake* en anglais.

Le travail, que nous présentons dans ce mémoire, s'inscrit dans une problématique de détection des contours d'une tumeur (qui représente l'objet à segmenter, dans ce cas) dans une image médicale, à l'aide des *snakes*. Pour se faire, nous l'avons organisé comme suit :

- Dans le premier chapitre, nous allons développer quelques généralités du traitement d'images, en se focalisant sur la détection des contours.
- Dans le deuxième chapitre, nous allons présenter une étude détaillée sur les contours actifs, en se basant sur le *snake* GVF.
- Dans le troisième chapitre, nous allons décrire, avec des illustrations, notre approche adoptée.
- Les différents tests effectués et les résultats obtenus seront présentés, avec interprétations, dans le quatrième chapitre.
- Ce mémoire s'achèvera par une conclusion générale permettant de rappeler le travail réalisé et d'éventuelles perspectives.

Chapitre I



Généralités

I.1 Introduction

Le traitement d'images constitue, actuellement, l'une des grandes orientations du traitement de l'information acquise à partir des capteurs, offrant des capacités de plus en plus importantes. L'essor des potentialités des calculateurs a permis d'envisager des procédures, de plus en plus complexes, pour mettre en évidence ce que l'œil peut voir.

Le traitement d'images désigne l'ensemble des méthodes dont l'objectif est, soit de transformer des images pour améliorer la surface en corrigeant les dégradations, soit d'extraire des informations qui permettent une interprétation visuelle ou automatique. C'est un domaine très vaste, qui trouve de plus en plus d'applications. Pour cela, nous allons donner des notions sur le traitement d'images, en commençant par rappeler, brièvement, les caractéristiques d'une image numérique. Ensuite, nous nous focaliserons sur les différents traitements que nous allons avoir besoin dans notre travail. Nous accentuons sur la détection des contours.

I.2 Caractéristiques d'une image numérique [1]

L'image numérique est un ensemble structuré d'informations, caractérisé par les paramètres ci-dessous.

I.2.1 Pixel

C'est une contraction de l'expression anglaise "*Picture elements*", qui signifie éléments d'image. Le pixel est le plus petit point de l'image. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression.

Dans le cas d'une image monochrome, chaque pixel est souvent codé sur un octet. La taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image. Dans une image couleur (R, V, B), un pixel est représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

I.2.2 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice. Le produit du nombre de lignes, de cette matrice, par le nombre de colonnes nous donne la dimension de l'image, qui est aussi le nombre total de pixel.

I.2.3 Résolution

Elle définit la clarté ou la finesse des détails reproduit dans l'image fictive par apport a l'image réelle.

La résolution est une grandeur qui mesure le rapport entre le nombre de pixel et la surface réelle de l'image représentée par ces pixels.

I.2.4 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de celle-ci. Pour un observateur lointain, le mot luminance est substitué à l'éclat d'un objet.

I.2.5 Contraste

C'est l'opposition entre deux régions d'une image, les régions sombre et les régions claires. Le contraste est défini en fonction des luminances de deux zones de l'image.

Si L_1 et L_2 sont les degrés de luminosité de deux zones voisines A_1 et A_2 d'une image, le contraste est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (I.2)$$

I.2.6 Contour [3]

Les contours représentent la frontière entre les objets de l'image, qui se traduit par une brusque variation du niveau de gris. Un contour peut être défini comme une marche d'escalier s'il est net, comme une rampe s'il est plus flou où comme un toit s'il s'agit d'une ligne sur un fond uniforme (Fig. I.1).

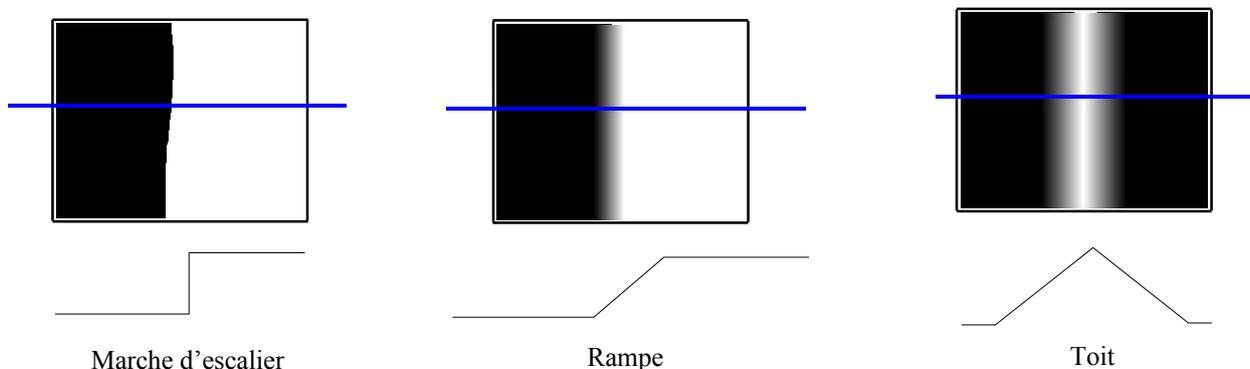


Fig. I.1 : les trois modèles du contour.

I.2.7 Texture [4]

La texture est une caractéristique importante de l'apparence des objets dans des scènes réelles. La comprendre est une partie essentielle de la compréhension de la vision humaine. Une texture représente, à une échelle donnée, le même aspect quelle que soit la zone observée. Dans ces conditions, on considère l'image comme la réalisation d'un processus stochastique local et stationnaire. C'est-à-dire que chaque pixel est caractérisé par un petit voisinage, et que cette caractérisation est la même pour tous les pixels de l'image.

On rencontre deux types de définitions de la texture : la première est *déterministe* et fait référence à une répétition spatiale d'un motif de base dans différentes directions. Cette approche structurelle correspond à une vision *macroscopique* des textures. La deuxième est *probabiliste* et cherche à caractériser l'aspect anarchique et homogène, qui ne comprend ni de motif localisable, ni de fréquence de répétition principale.

I.2.8 Bruit

Un bruit (ou parasite), dans une image, est considéré comme un phénomène de brusque variation d'intensité d'un pixel par rapport à ses voisins. Il peut provenir de l'éclairage des dispositifs optiques et électroniques du capteur.

I.3 Histogramme [1]

I.3.1 Définition

Un histogramme est un ensemble de données statistiques permettant de représenter la distribution des intensités lumineuses des pixels d'une image (Fig. I.2), c'est-à-dire le nombre de pixels pour chaque niveau de gris. Pour une image en couleur, il est possible soit de faire un histogramme par composante (rouge, vert et bleue), soit de faire l'histogramme de la moyenne des trois composantes pour chaque pixel.

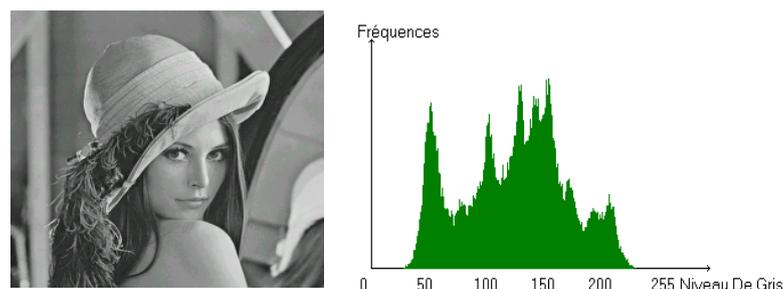


Fig. I.2 : image en niveau de gris et son histogramme.

I.3.2 Types d'histogrammes

On peut distinguer trois types d'histogrammes d'images : Unimodal, Bimodal, et Multimodal :

- ✓ **L'histogramme Unimodal** : il n'a qu'un seul pic qui représente soit un objet, soit un fond (Fig. I.3.(a)).
- ✓ **L'histogramme Bimodal** : il est formé de deux modes séparés (deux pics séparés par une vallée), qui représentent un objet sur un fond (Fig. I.3.(b)).
- ✓ **L'histogramme Multimodal** : il est formé de plusieurs modes séparés (plusieurs pics séparés par plusieurs vallées). Il nous renseigne sur la présence de plusieurs objets (Fig. I.3.(c)).

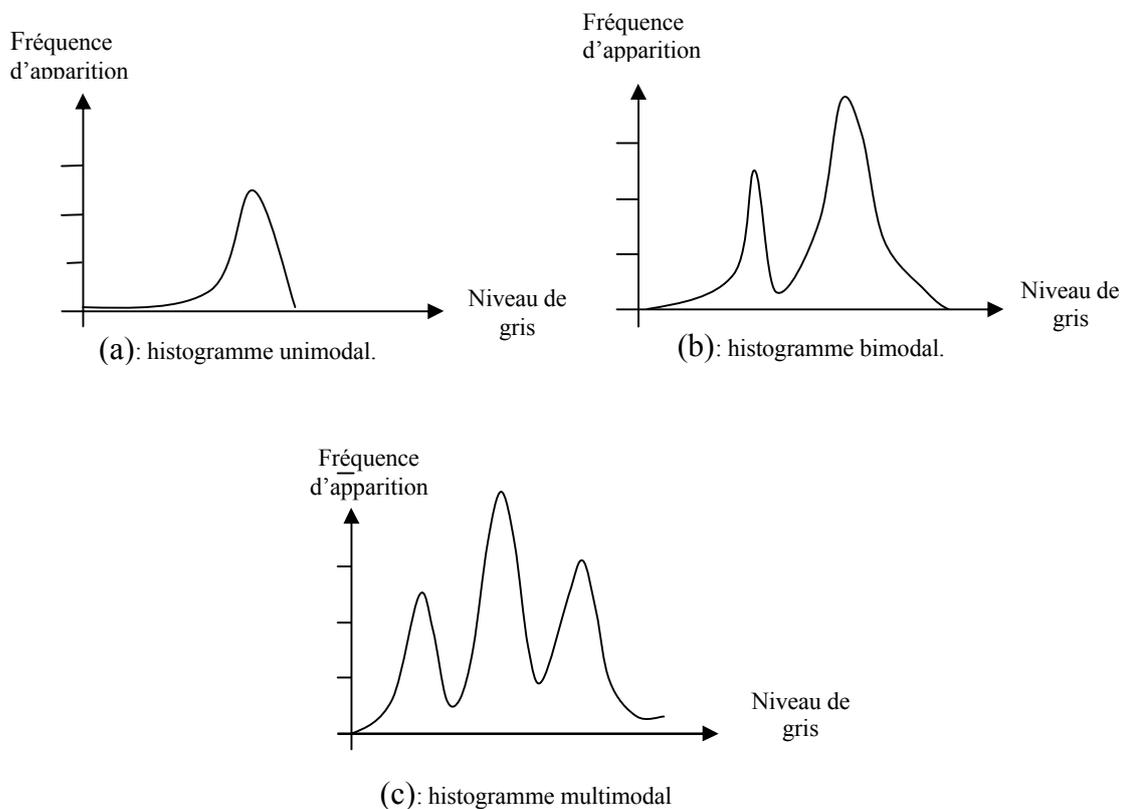


Fig. I.3 : types d'histogrammes.

I.3.3 Les différentes opérations appliquées sur l'histogramme

a) Modification d'histogramme

L'histogramme est un outil qui permet d'étudier la répartition des composantes d'une image. Sa modification est, généralement, représentée sur une courbe appelée courbe tonale

indiquant la modification globale des composantes de l'image, avec en abscisses les valeurs initiales et en ordonnées les valeurs après modification.

b) Egalisation d'histogramme

L'égalisation d'histogramme est un outil qui permet d'améliorer certaines images de mauvaise qualité (mauvais contraste, images trop sombres, ou trop claires, mauvaise répartition des niveaux d'intensité). Il s'agit d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel, pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image (Fig. I.4).

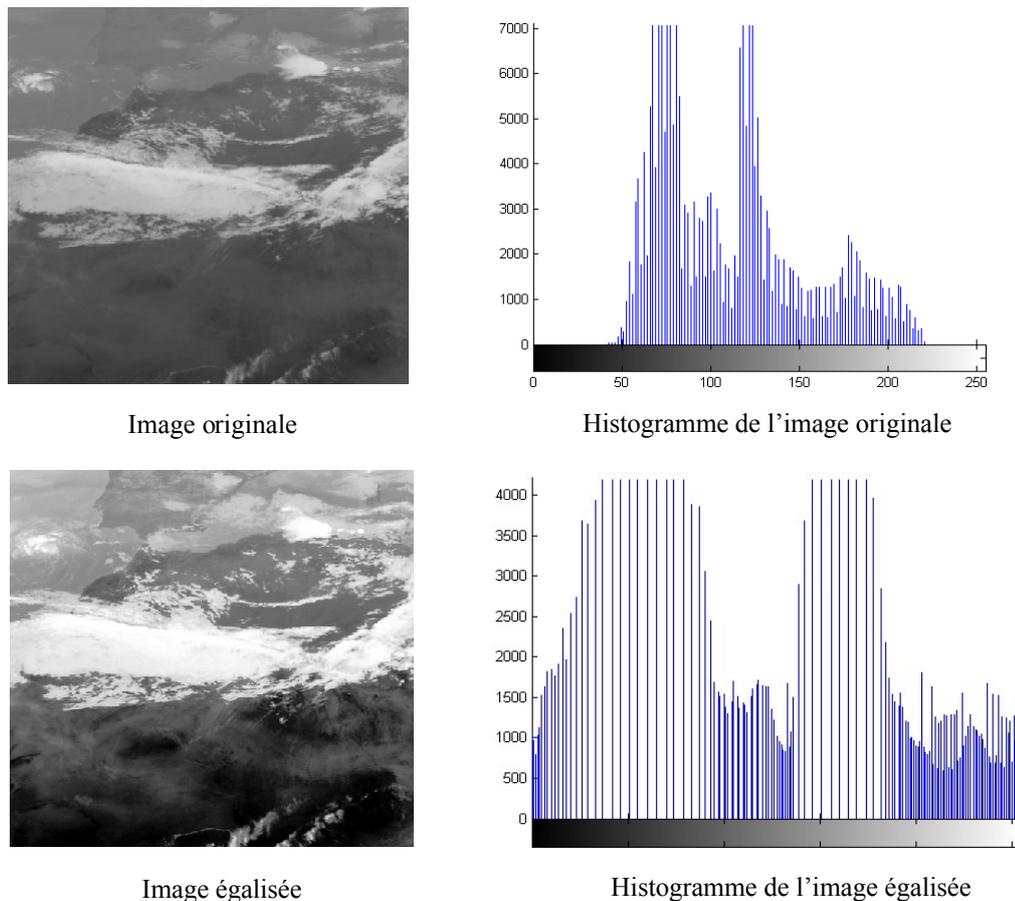


Fig. I.4 : exemple d'image avant et après égalisation de son histogramme.

c) Étirement d'histogramme

L'étirement d'histogramme (aussi appelé *linéarisation d'histogramme*), consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. L'objectif de cette opération est de modifier l'histogramme de telle manière à répartir, au mieux, les intensités sur l'échelle des valeurs disponibles. Ceci revient à étendre l'histogramme afin que la valeur d'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale.

d) Inversion d'images (négatif)

L'opération d'inversion consiste, comme son nom l'indique, à inverser les valeurs des pixels par rapport à la moyenne des valeurs possibles. Le résultat obtenu est appelé *négatif* (Fig. I.5).

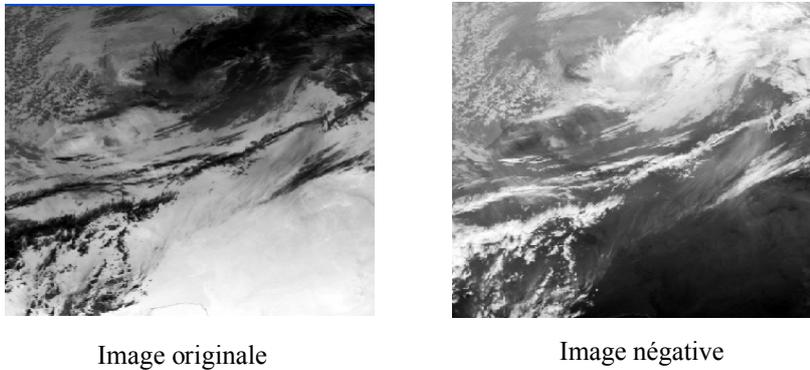


Fig. I.5 : exemple d'image et de son négatif.

I.4 Le seuillage [5]

Le seuillage a pour objectif la simplification de la représentation de l'image. Il consiste à transformer l'image codée sur 6,8 ou 16 bits en une image binaire, où les pixels à 1 correspondent aux objets et les pixels à 0 au fond de l'image.

L'opération de seuillage permet de mettre en évidence des formes ou des objets, selon les parties qui intéressent l'opérateur. Ceci se fait par la détermination d'un seuil S d'intensité, à base duquel les pixels seront séparés en deux classes : ceux appartenant au fond, et ceux appartenant aux objets (Fig. I.6).

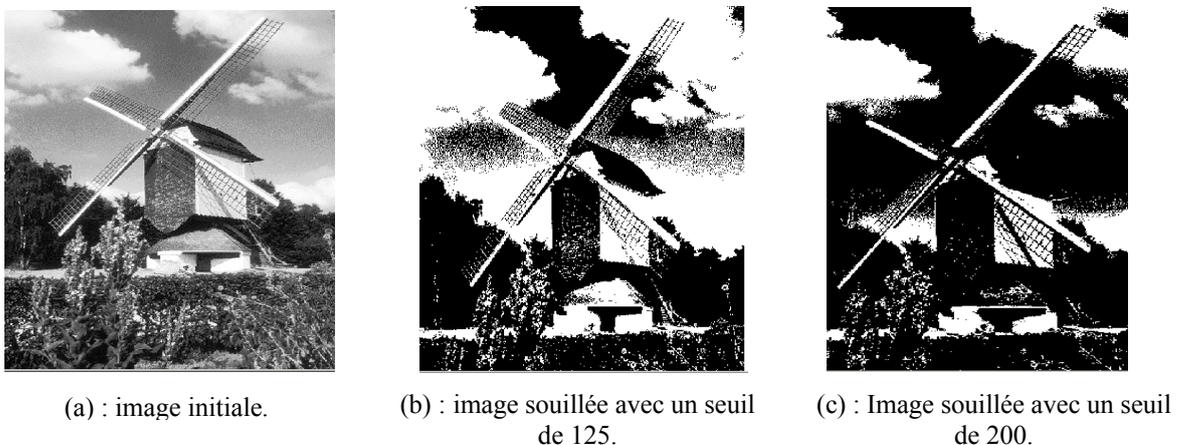


Fig. I.6 : exemple de seuillage d'image.

Cependant, la manière dont on détermine le seuil S est l'objet de plusieurs méthodes qui se scindent, essentiellement, en deux grandes familles :

- Les méthodes de seuillage fondées sur l'analyse de l'histogramme.
- Les méthodes des seuillages adaptatifs.

I-4-1 Le seuillage fondé sur l'analyse de l'histogramme

L'algorithme de seuillage va déterminer un seuil ou plusieurs seuils, cela va dépendre de la nature de l'image et de son histogramme associé. En effet, si celui-ci correspond à une image bimodale, l'algorithme trouvera un seul seuil. Si celui-ci correspond à une image multimodale, l'algorithme trouvera plusieurs seuils [6].

I.4.2 Le seuillage adaptatif (local)

Le seuillage adaptatif est un type de seuillage paramétrique, c'est-à-dire qu'il tient compte des variations d'éclairements présents dans l'image. En fait, ce type d'opération fait intervenir les plus proches voisins d'un pixel. On doit écrire, en fait, une équation de type $ng = I(x, y)$ pour obtenir le niveau de gris (ng) d'un pixel par rapport à ses coordonnées (x, y) dans l'image.

Le choix d'un seuil, variant en fonction de la région de l'image, serait plus adéquat. Cela nécessite, donc, une partition de l'image en sous régions et la détermination d'un seuil pour chaque sous région.

L'ensemble des seuils $\{s_1, s_2, \dots, s_n\}$ serait tel que tous les points image satisfont la condition suivante $I(x, y) \in \{s_i, s_{i+1}\}$, i étant la $i^{\text{ème}}$ région et $i = 1 \dots n$ [6].

I.4.3 La technique globale [7]

Elle consiste à trouver un seuil S divisant l'histogramme en deux zones distinctes de niveaux de gris, en vue de déterminer les frontières entre les objets et le fond. Ce seuil est déterminé en considérant tous les pixels de l'image en une fois. Ensuite une comparaison de chaque pixel au seuil est faite.

I.5 Filtrage

I.5.1 Définition

Le filtrage est une opération fondamentale en traitement d'image. Il permet d'améliorer la perception de certains détails, de réduire le bruit et de compenser les défauts du

capteur. On note trois catégories de filtrages : le filtrage linéaire, le filtrage adaptatif et le filtrage morphologique.

I.5.2 Filtrage linéaire [9]

Filtrer une image revient à effectuer le produit de convolution entre $I(i, j)$ et la réponse impulsionnelle $F(i, j)$ du filtre.

Dans le cas continue l'image filtrée est donnée par :

$$I'(i, j) = (F * I)(i, j) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (F(i_1, j_1)) I(i - i_1, j - j_1) di_1 dj_1 \quad (I.3)$$

Dans le cas discret, les domaines de définition de I et F sont bornés. Le domaine de F est $\left[-\frac{k}{2}, +\frac{k}{2}\right]$ avec k qui est inférieur à la taille de l'image. La convolution s'écrit alors :

$$I'(i, j) = (F * I)(i, j) = \sum_{i'=-k/2}^{+k/2} \sum_{j'=-k/2}^{+k/2} F(i', j') I(i - i', j - j') \quad (I.4)$$

Le plus grand avantage de ces filtres est leur facilité de conception et d'implémentation. Mais, ils ne peuvent pas être utilisés pour des travaux trop fins, car la détérioration des contours qu'ils introduisent, par exemple, empêchera une segmentation fine des images. Les filtres linéaires, fréquemment utilisés, sont les filtres passe-bas, passe-haut, passe-bande et directionnels.

I.5.3 Filtrage adaptatif ou non linéaire

Les filtres non linéaires sont des filtres régis par une fonction non linéaire. Leur réponse n'est plus une combinaison linéaire, contrairement, aux filtres linéaires. Les filtres adaptatif qui utilise de l'information à *priori* sur l'image en question, qui ce traduit généralement par un jeu de paramètres. On y trouve principalement les filtres d'ordre et les filtres de position.

I.5.4 Filtrage morphologique [10]

Une autre approche pour le filtrage consiste à considérer l'image, non comme une fonction bidirectionnelle mais plutôt comme un espace de représentation d'objets ayant des propriétés ensemblistes. Cette approche, appelée morphologie mathématique, permet de conserver les propriétés algébriques, géométriques, topologiques et ensemblistes. Ce type de filtre est souvent appliqué après binarisation de l'image.

Les algorithmes de morphologie mathématique permettent de travailler sur les images binaires, permettant la mise en évidence de leurs propriétés particulières. Ces algorithmes sont basés sur l'utilisation d' « éléments structurants », c'est à dire des configurations élémentaires de pixels à rechercher dans l'image. Parmi ces opérateurs nous citons : l'érosion, la dilatation, la fermeture et l'ouverture (Fig. I.7).

L'érosion : de I par l'élément structurant B est défini de la manière suivante :

$$E_B(x, y) = \inf_{i, j \in B} I(x + i, y + j) \tag{I.5}$$

Sur les images au niveau de gris, cet opérateur permet de supprimer toutes les hautes fréquences claires de l'image. Sur les images binaires, il permet de cacher les petits points blancs.

La dilatation : de I par l'élément structurant B est défini de la manière suivante :

$$D_B(x, y) = \sup_{i, j \in B} I(x + i, y + j) \tag{I.6}$$

Sur les images au niveau de gris, ce filtre permet de supprimer toutes les hautes fréquences sombres de l'image. Sur les images binaires, il permet de cacher les petits points noirs.

Dans l'exemple suivant, nous avons appliqué les opérateurs de dilatation et d'érosion sur la forme initiale avec les trois éléments structurants de base.

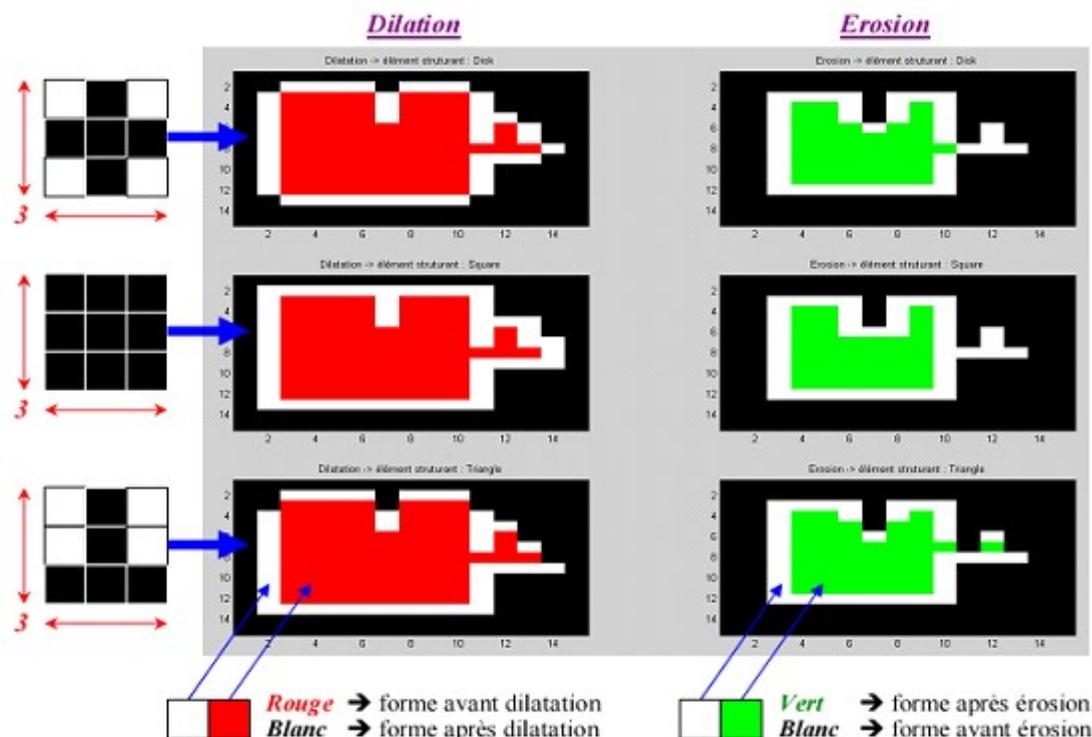


Fig. I.7 : opérateurs morphologiques d'érosion et de dilatation avec les 3 éléments structurants.

Après érosion, les objets de taille inférieure à celle de l'élément structurant vont disparaître, et les autres seront "amputés" d'une partie correspondante à la taille de l'élément structurant. S'il existe des trous dans les objets, c'est à dire des "morceaux" de fond à l'intérieur des objets, ils seront accentués, et les objets reliés entre eux vont être séparés. Pour la dilatation, c'est le contraire qui va se passer (Fig. I.8).

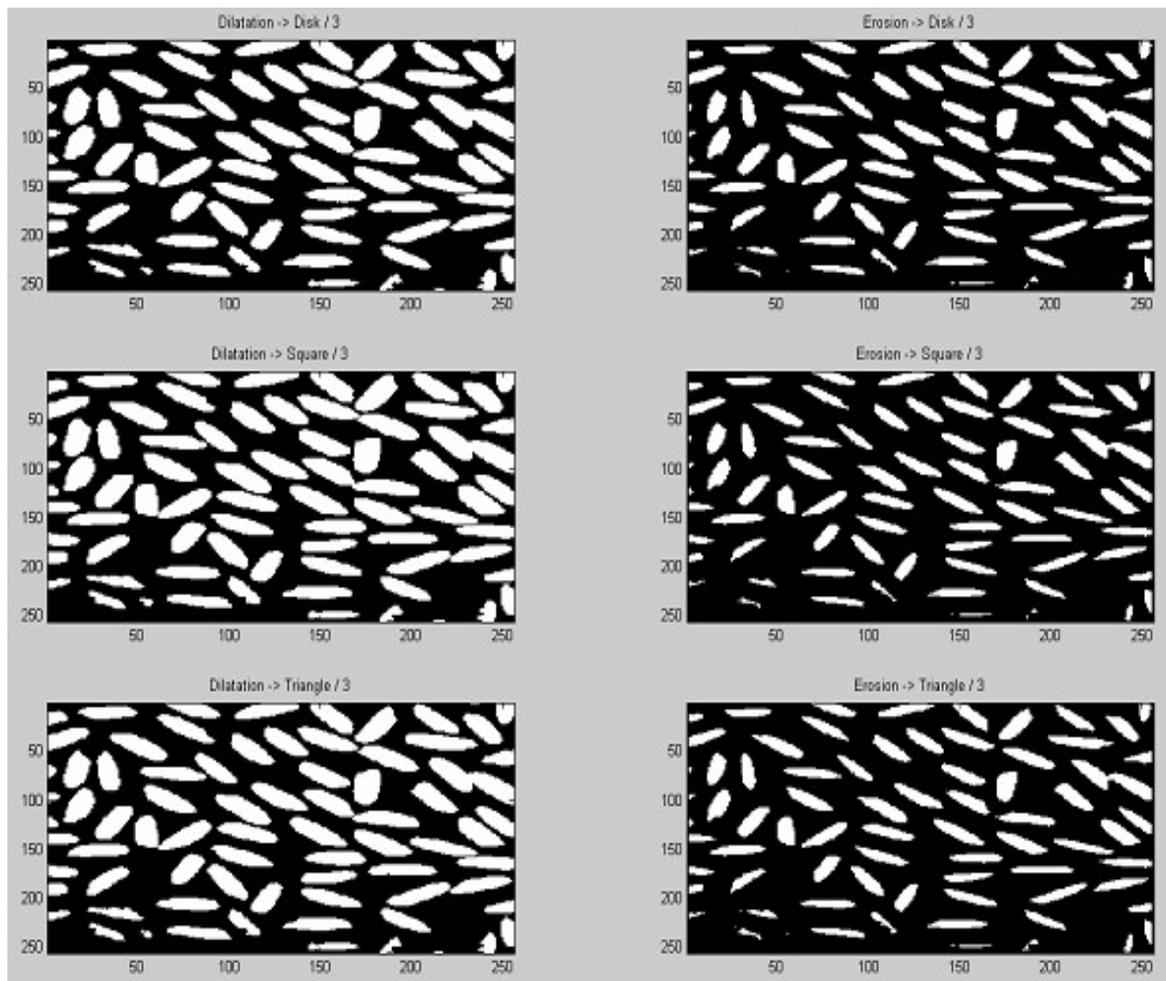


Fig. I.8 : opérateurs morphologiques d'érosion et de dilataion avec les 3 éléments structurant de taille 3 x 3 : image « Rice.tif » binarisée.

L'ouverture : est obtenue en effectuant sur l'image une érosion suivit d'une dilataion comme suit :

$$O_B(I) = D_B(E_B(I)) \quad (I.7)$$

La fermeture : est obtenue on effectuant sur l'image une dilataion suivit d'une érosion comme suit :

$$F_B(I) = E_B(D_B(I)) \quad (I.8)$$

Comme le montre les images suivantes, l'ouverture a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant (Fig. I.9). La fermeture a pour propriété de combler tout ce qui est de taille inférieure à l'élément structurant.

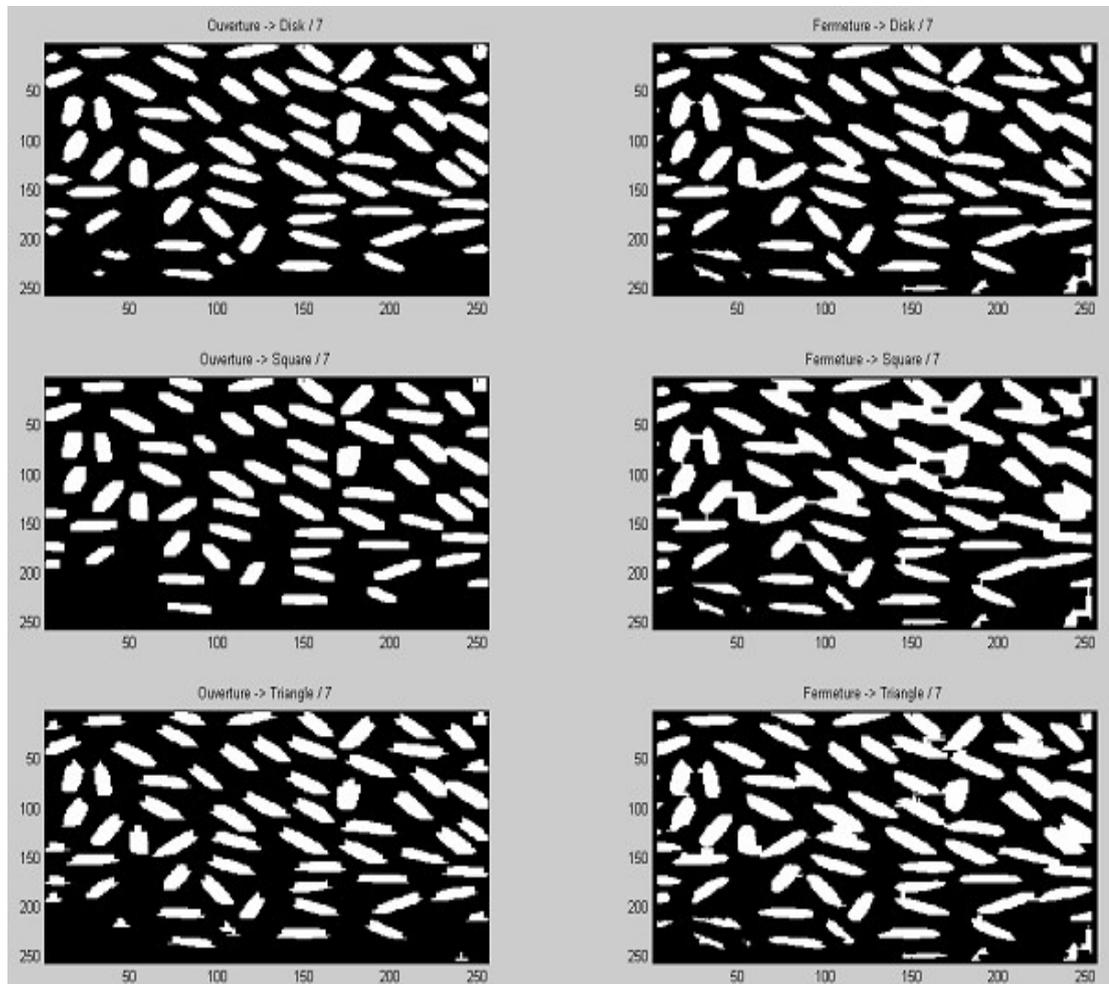


Fig. I.9 : opérateurs morphologiques d'ouverture et de fermeture avec les 3 éléments structurants de taille 7 x 7 : image « Rice.tif » binéarisée.

I.6 La segmentation

I.6.1 Définition [8]

Segmenter une image, c'est effectuer une partition de cette image en régions, telle que chacune d'elles possède, au moins, une caractéristique que ne possède pas les régions voisines. Le but de la segmentation est de fournir une description de l'image en liste de régions caractérisées par des propriétés qui les distinguent. Plusieurs approches (algorithmes) de segmentation existent, selon la diversité des propriétés recherchées, les problèmes

rencontrés et la manière d'opérer pour le regroupement des pixels. On distingue, généralement, trois approches : classification des pixels, approche régions et approche frontières.

I.6.2 Approches de la segmentation

I.6.2.1 Approche par classification des pixels [1]

La classification des pixels se fait sur la base de leur ressemblance. On définit un ensemble fini de classes de pixels et on affecte chaque pixel de l'image à l'une de ces classes. Des régions se forment par le regroupement des pixels connectés par la propriété d'appartenance à une même classe.

I.6.2.2 Approche régions [1]

Dans cette approche, les pixels sont traités en groupes, contrairement à l'approche par classification des pixels où ils sont traités individuellement. La segmentation est réalisée par étapes. Le processus est initialisé par la définition d'une partition de base, puis par étapes successives ces images sont remodelées. Ces méthodes sont basées sur la construction des régions suivant un critère d'homogénéité.

I.6.2.3 Approche frontières [8]

Cette approche est une technique non contextuelle, qui ignore les relations pouvant existées entre les régions de l'image. Les pixels sont regroupés en fonction d'un attribut global. Elle comprend les techniques de détection de contours. Cependant, les contours obtenus ne conduisent pas, toujours, directement à la partition recherchée. En effet, les pixels contours mis en évidence pour une forme, généralement, ne sont pas connexes. Il faut alors appliquer des algorithmes de fermeture de contours. Ce n'est qu'après fermeture des contours que les régions apparaissent, définies par l'intérieur des contours.

I.7 Détection des contours

Un contour est une brusque variation de niveau de gris dans une image. Il est aussi définit comme la frontière entre deux régions dissemblables dans une image. Les contours sont, aussi, les lieux de variations significatives de l'information.

L'extraction de contours a pour but d'identifier les caractéristiques significatives de l'image. Elle se base sur la distribution des niveaux de gris des pixels. Le principe de cette méthode est, essentiellement, d'effacer tous les motifs à faible variation de niveau de gris (ou

de couleur) pour ne conserver que les lignes de séparation entre régions homogènes. Pour cela, différentes approches sont utilisées, nous citerons les approches dérivatives, les méthodes surfaciques et, plus récemment, les contours actifs.

I.7.1 Les approches dérivatives [9]

Les approches dérivatives regroupent les différents opérateurs basés sur le calcul du gradient, les filtres optimaux et le *Laplacien*.

I.7.1.1 Approche gradient

Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son module et sa direction. Le module est directement lié à la quantité de variation locale des niveaux de gris. La direction du gradient est orthogonale à la frontière qui passe au point considéré (Fig. I.10) et s'oriente de la partie claire vers la partie foncée.

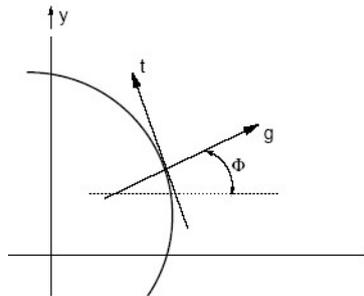


Fig. I.10 : gradient d'un point donné appartenant à une frontière donnée.

En considérant $f(x, y)$ la fonction de niveau de gris de l'image, le gradient permet de mesurer les vitesses de changement des valeurs de cette fonction par rapport aux distances dans la direction x et y , par le maximum de la dérivée première (Fig. I.11(b)) ou par le passage à zéro de la dérivée seconde (Fig. I.11(c)).

Le gradient, en un point de coordonnées (x, y) , est défini par :

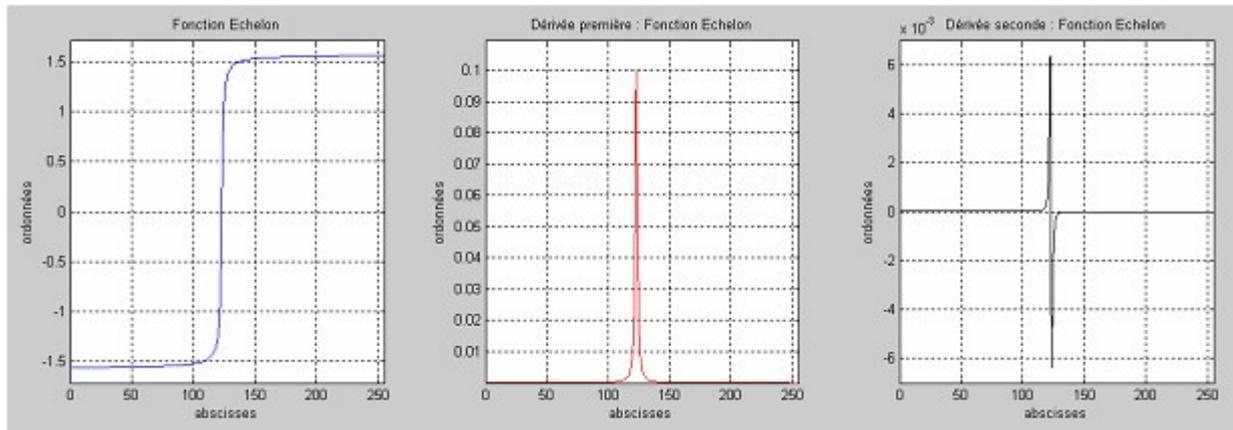
$$\vec{G} = \vec{\nabla}f = \begin{bmatrix} \nabla f_x \\ \nabla f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} \quad (\text{I.9})$$

Le module est donné par :

$$G = |\vec{\nabla}f| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (\text{I.10})$$

Tandis que son orientation est définie par :

$$|\theta(x, y)| = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (\text{I.11})$$



(a) signal initial

(b) dérivée première

(c) dérivée seconde

Fig. I.11 : extraction des contours par la première et la deuxième dérivée.

Le calcul des dérivées peut être réalisé de différentes méthodes soit par le calcul direct ou par différents masques qui seront détaillés ci-après.

a) Calcul direct

Il est réalisé de façon simple :

$$G_x = f(x + 1, y) - f(x, y) \quad (\text{I.12})$$

$$G_y = f(x, y + 1) - f(x, y)$$

Ce qui correspond à une convolution avec les masques :

$$G_x = [-1 \quad 1] \quad \text{et} \quad G_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad (\text{I.13})$$

b) Opérateur de Roberts :

Cet opérateur permet de calculer le gradient bidimensionnel, d'une image, de manière simple et rapide. Il amplifie les zones où la norme du gradient spatial est importante, qui correspond souvent aux contours. L'opérateur de Roberts est, généralement, très sensible aux bruits. Ces effets peuvent être supprimés si on effectue un lissage préalable sur toute l'image. Les masques de Roberts sont les suivants :

1	0
0	-1

Filtre diagonal

0	1
-1	0

Filtre anti-diagonal

Sur la figure ci-dessous, nous donnons un exemple d'image après lui avoir appliqué l'opérateur de *Roberts* (Fig. I.12).



(a) Image originale



(b) Image des contours.

Fig. I.12 : utilisation de l'opérateur de *Roberts* pour la détection de contours.

c) Opérateur de *Sobel* :

Pour cet opérateur, nous utilisons les 2 masques suivants, pour calculer le gradient :

1	0	-1
2	0	-2
1	0	-1

 h_x

1	2	1
0	0	0
-1	-2	-1

 h_y

Pour illustrer le résultat de cette opération, nous donnons un exemple sur la figure (I.13)



(a) Image originale



(b) Image des contours.

Fig. I.13 : utilisation de l'opérateur de *Sobel* pour la détection de contours.

d) Opérateur de Prewitt

L'opérateur de *Prewitt* combine à la fois un filtrage et une dérivée. Il est donc moins sensible aux bruits que le calcul direct des dérivées. Les masques dérivatifs sont dans ce cas :

$$\begin{array}{ccc}
 \begin{array}{|c|c|c|}
 \hline
 1 & 0 & -1 \\
 \hline
 1 & 0 & -1 \\
 \hline
 1 & 0 & -1 \\
 \hline
 \end{array} & & \begin{array}{|c|c|c|}
 \hline
 1 & 1 & 1 \\
 \hline
 0 & 0 & 0 \\
 \hline
 -1 & -1 & -1 \\
 \hline
 \end{array} \\
 h_x & & h_y
 \end{array}$$

Le résultat de ce masque sur l'image *Lena* est illustré sur la figure (I.14) ci-dessous.

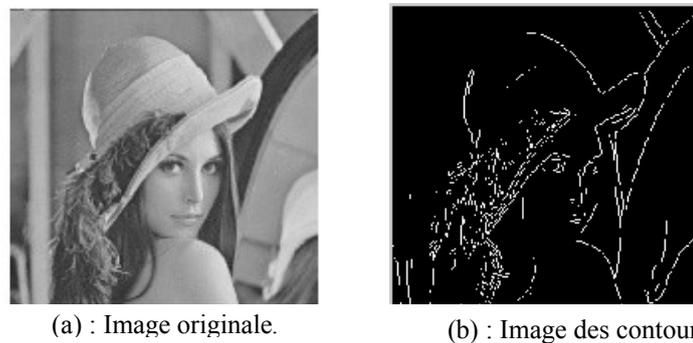


Fig. I.14 : utilisation de l'opérateur de *Prewitt* pour la détection de contours.

Les opérateurs de *Sobel* et de *Prewitt* permettent d'estimer, localement, la norme du gradient spatial bidimensionnel, d'une image en niveau de gris. Ils amplifient les régions de fortes variations locales d'intensité correspondante aux contours.

e) Masque dérivé de gaussienne

Dans ce cas, la dérivée peut être réalisée grâce à un masque gaussien G . Ainsi, le calcul de la dérivée en x se définit comme suit :

$$f_x = \frac{\partial(f * G)}{\partial x} = f * \frac{\partial G}{\partial x} \quad (\text{I.14})$$

Plutôt que de calculer deux produits de convolution sur toute l'image (coûteux en temps de calcul), on calcule le produit de convolution de l'image et de la dérivée en x du filtre. On aura donc :

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \quad (\text{I.15})$$

Soit

$$\frac{\partial G(x, y)}{\partial x} = -\frac{x}{2\pi\sigma^4} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \quad (\text{I.16})$$

Le paramètre σ^2 règle le degré de lissage.

Le même raisonnement sera tenu pour la dérivée en y .

I.7.1.2 Les filtres optimaux

Dans ces approches, qui font partie des méthodes dérivatives, un contour est modélisé par un échelon d'amplitude U_0 , noyé dans un bruit blanc. Il s'agit alors de calculer un filtre optimal pour détecter et localiser la transition.

L'approche de *Canny* consiste, dans le cas monodimensionnel, à trouver le filtre optimal de réponse impulsionnelle $h(x)$, satisfaisant les trois contraintes suivantes : une bonne détection, une bonne localisation avec précision et une réponse unique. *Dérêche* a utilisé la même démarche que *Canny*, pour chercher une réalisation de l'opérateur sous la forme d'un filtre à réponse impulsionnelle infinie. Dans le cas bidimensionnel, le filtre utilisé est la combinaison de deux filtres monodimensionnels dans les directions x et y . Un exemple d'application du filtre de *Canny* est représenté sur la figure (I.15).

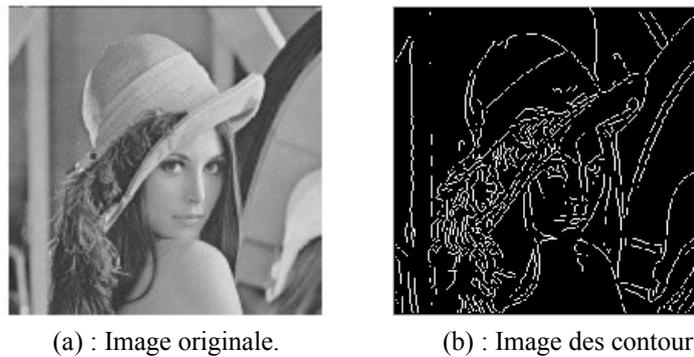


Fig. I.15 : utilisation de l'opérateur de Canny pour la détection de contours.

I.7.1.3 Approche *Laplacien*

La recherche des contours dans l'image peut, aussi, être réalisée en cherchant les passages par zéro du *Laplacien*. Celui-ci peut être estimé à une convolution avec les masques :

0	1	0
1	-4	1
0	1	0

M_1

-1	-1	-1
-1	8	-1
-1	-1	-1

M_2

1	-2	1
-2	1	-2
1	-2	1

M_3

Un exemple pour illustrer l'approche *Laplacien* est représenté sur la figure (I.16).

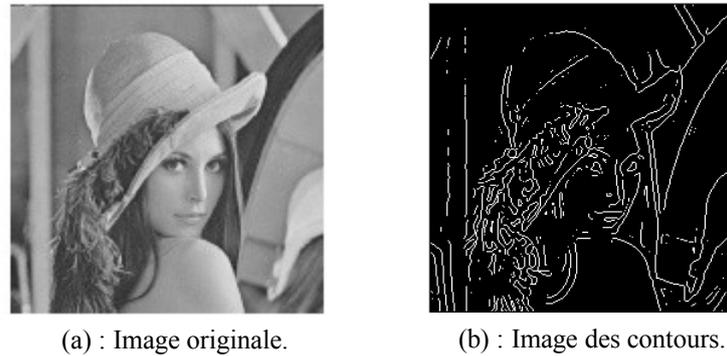


Fig. I.16 : utilisation de l'opérateur *Laplacien* pour la détection de contours.

Dans la figure (I.17), nous avons superposé sur l'image *Rice*, les différents contours détectés avec quelques approches.

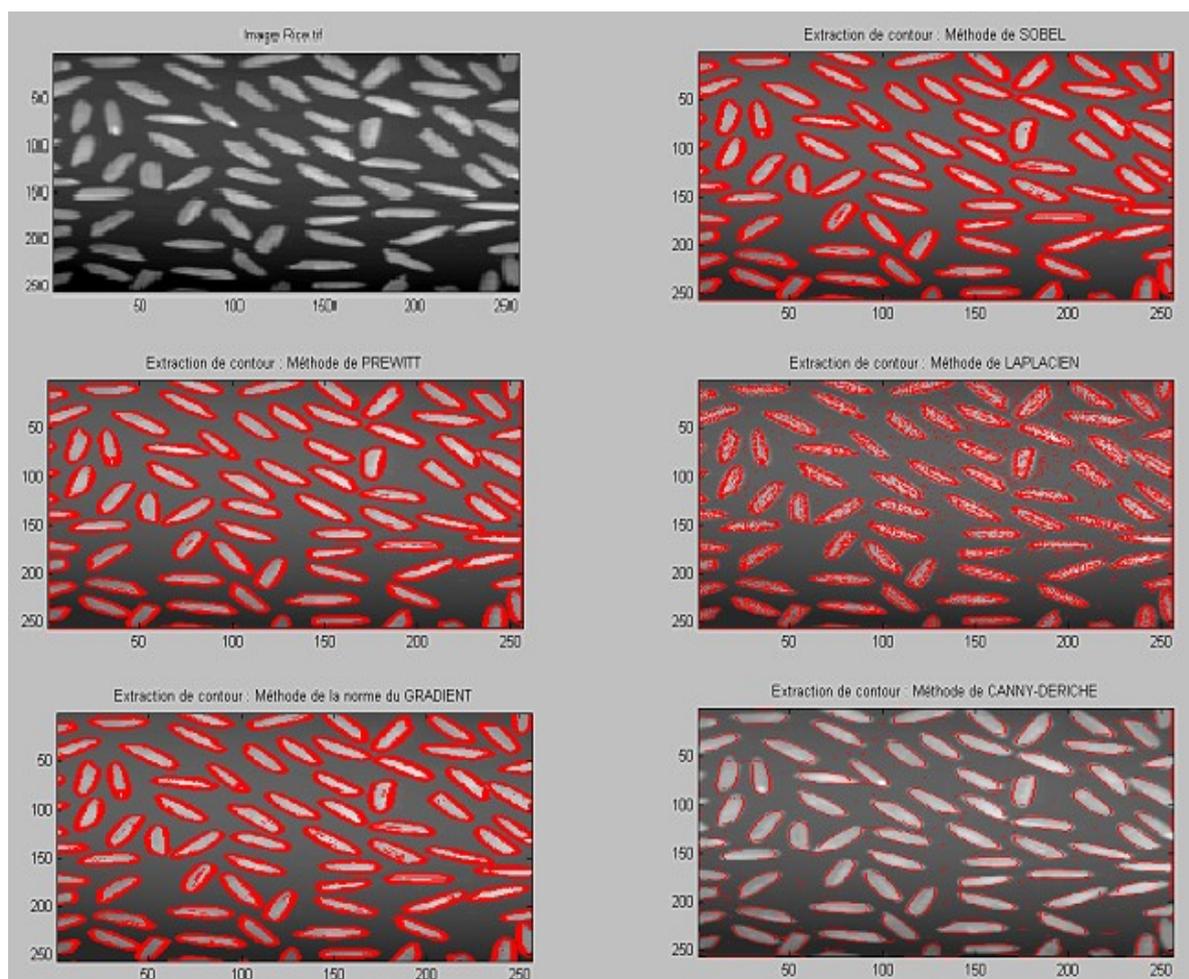


Fig. I.17 : combinaison des contours détectés par quelques approches dérivatives, avec l'image Rice.

I.7.2 Les méthodes surfaciques [10]

En considérant la fonction d'intensité d'une image comme une surface, le contour est considéré comme une courbure dans un espace à deux dimensions. Puisque cette méthode prend en compte les propriétés géométriques des contours à deux dimensions, elle peut donner la position précise de ses dernières.

I.7.3 Les contours actifs [11]

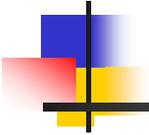
Un contour actif est un ensemble de points qu'on va tenter de déplacer, pour leur faire épouser une forme. L'idée de cette méthode est de déplacer les points pour les rapprocher des zones de fort gradient, tout en conservant des caractéristiques comme la courbure du contour ou la répartition des points sur le contour, d'autres contraintes liées à la disposition des points.

I.8 Conclusion

Le traitement d'image est né pour nouveaux besoins, que les nouvelles technologies ont engendrés. Il s'est, grandement, appuyé sur ces mêmes nouvelles technologies pour se développer très rapidement.

Nous avons présenté dans ce chapitre quelques notions essentielles sur le traitement d'images. Pour la détection de contour, nous avons présenté quelques approches. Les résultats de ces opérations sont peu concluants. Nous ne pouvons effectuer une segmentation à base de ses contours seuls, car ses derniers sont le plus souvent discontinus et incomplets.

Pour remédier à ce problème, de nombreuses méthodes ont été proposées. La segmentation par les contours actifs est l'une de ces méthodes les plus récentes, et qui fera l'objet d'étude dans le chapitre suivant.



Chapitre II :

Contours actifs

II.1 Introduction

Depuis les travaux de *Kass, Witkin et Terzopoulos* sur les *snakes*, les contours actifs ont connu un progrès important en imagerie.

Un contour actif est une forme continue (fermé ou non) capable d'évoluer, plus exactement de se déformer, de manière à venir se mettre en correspondance avec les données traitées. Dans une image 2D, il est représenté par un contour et dans une image 3D par une surface.

Différentes classes de contours actifs sont développés ces dernières années. Pour cela, nous allons dans ce chapitre faire une étude de ces contours en commençant par leur principe jusqu'à leurs différents types.

II.2 Principe des contours actifs [12]

Le principe des contours actifs est de faire évoluer une courbe initiale, définie par l'utilisateur, en direction des bords des objets à détecter. L'équation d'évolution générale régissant les contours actifs est la suivante :

$$\frac{\partial C(s,t)}{\partial t} = V(s,t) \quad (\text{II.1})$$

Avec: $C(s, 0) = C_0(s)$, qui est un contour initial définie par l'utilisateur, $C(s, t)$ est une courbe qui évolue avec la vitesse $V(s, t)$ (Fig. II.1).

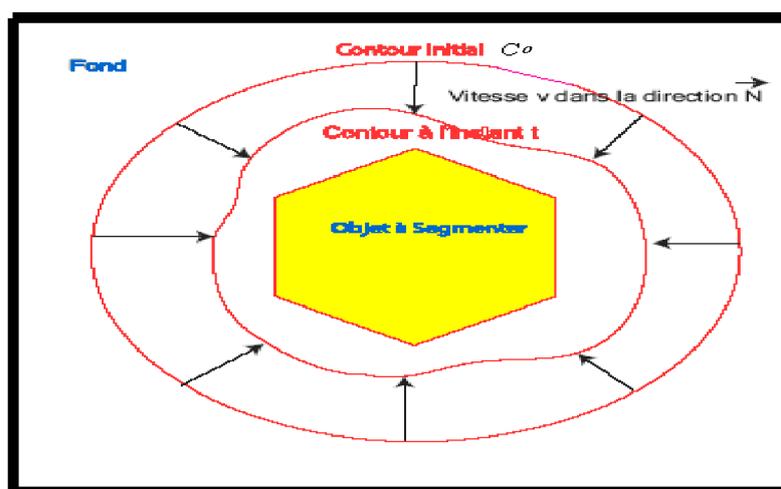


Fig. II.1 : évolution du contour actif $C(t)$ en direction des bords avec une vitesse V .

II.3 Formulation mathématique

Le contour actif peut être décrit par une courbe C , fonction de temps t et de l'abscisse curviligne s , par :

$$\begin{aligned} v : \Omega = [0, 1] &\longrightarrow \mathbb{R}^2 \\ S &\longrightarrow C(s) \equiv \{v(s)\} = (x(s, t), y(s, t)) \end{aligned} \quad (\text{II.2})$$

Avec : $v(s, t)$ est la position d'un point de la courbe à un instant t et (x, y) sont les coordonnées cartésiennes d'un point de l'image (Fig. II.2).

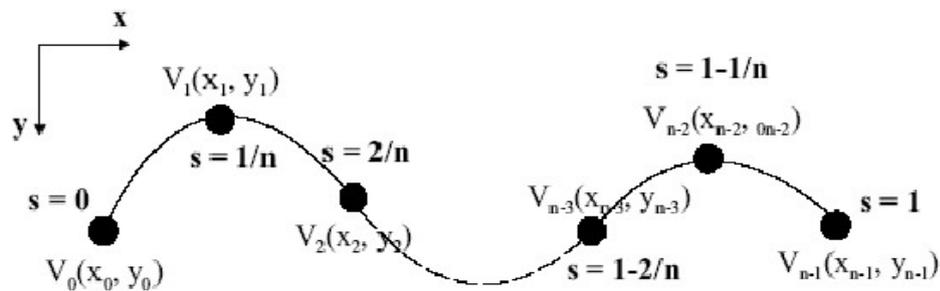


Fig. II.2 : coordonnées cartésiennes et abscisses curvilignes pour un *snake* de n points.

Le calcul de l'énergie est, ensuite, défini de façon à évoluer suivant deux contraintes :

$$E(v) = E(v)_{\text{int.}} + E(v)_{\text{ext}} \quad (\text{II.3})$$

Avec: $E(v)_{\text{int.}}$: L'énergie interne du contour.

$E(v)_{\text{ext}}$: L'énergie externe du contour.

Lorsque le contour actif coïncide avec le contour à détecter, on atteint le minimum de cette énergie.

II.3.1 Energie interne

L'énergie interne détermine la régularité et la forme lisse du contour. Elle sert, aussi, à maintenir la cohérence et la cohésion des points et la raideur de la courbe. L'énergie interne modélise, entre autre, la tension. Elle est définie par [8] :

$$E_{int} = \frac{1}{2} \int_0^1 \left[\alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{ds^2} \right|^2 \right] ds \quad (\text{II.4})$$

Avec v : la position d'un point de la courbe.

$\frac{dv}{ds}$: la vitesse d'évolution du point.

$\frac{d^2v}{ds^2}$: son accélération.

Cette énergie se compose de 2 termes : un terme du 1^{er} ordre, contrôlé par $\alpha(s)$, qui représente l'élasticité du contour, et un autre terme du second ordre, contrôlé par $\beta(s)$, qui représente la rigidité du contour.

II.3.2 Energie externe

L'énergie externe (ou de contraintes) est définie par l'utilisateur. Elle permet de déterminer les critères d'évolution du contour en fonction de l'image $I(x, y)$. Elle correspond à l'adéquation aux données. C'est l'énergie potentielle liée à l'image. Elle est donnée par l'expression suivante :

$$E_{ext} = \int_0^1 P(v(s)) ds \quad (\text{II.5})$$

Où $P(\cdot)$ est une fonction d'attraction de contour définie sur le plan image. Généralement, elle est donnée par l'expression suivante [8]:

$$P(x, y) = -\lambda |\nabla G_\sigma * I(x, y)| \quad (\text{II.6})$$

Où G_σ est le filtre de Gauss de variance, σ , λ une constante convenablement choisie et ∇ est

l'opérateur gradient défini par $\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}$.

II.3.3 Processus de minimisation

L'évolution du contour actif est régit par la minimisation de son énergie totale E_{snake} :

$$E_{snake} = \frac{1}{2} \int_0^1 \left[\alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{ds^2} \right|^2 \right] ds - \lambda \int_0^1 |\nabla G_\sigma * I(x, y)| ds \quad (\text{II.7})$$

Avec un ensemble donné de paramètres β et α , cette équation intégrale est résolue par plusieurs méthodes. Des méthodes appelées méthodes locales, cherchant un minimum local, et celles appelées méthodes globales qui cherchent un minimum global.

II.3.3.1 Méthodes d'optimisation locale [12]

Il existe plusieurs méthodes d'optimisation locale. Parmi ces méthodes on peut citer celles de descente du gradient, les méthodes variationnelles et l'algorithme glouton.

➤ *Méthode de la descente du gradient*

La méthode de descente du gradient, est caractérisée par l'évolution d'une solution initiale v_0 (courbe par exemple), dans la direction de la plus forte pente (énergie par exemple). La solution évolue alors selon l'équation :

$$v_{i+1} = v_i - \Delta t \nabla E_p(v_i) \quad (\text{II.8})$$

Où : Δt est le pas du gradient et $\nabla E_p(v_i)$ est le gradient de l'énergie potentielle.

Cette méthode fait converger vers un minimum local de l'énergie. La solution initiale est donc primordiale. D'autre part, cela fait évoluer le modèle à pas constant.

➤ *Méthodes itératives*

Dites aussi méthodes variationnelles. C'est la classe de méthodes d'optimisation la plus connue et la plus utilisée dans les modèles déformables. Elle consiste à trouver une courbe qui minimise la fonctionnelle d'énergie décrite dans l'équation (II.7). La courbe résolvant ce problème doit satisfaire l'équation d'Euler-Lagrange suivante :

$$\frac{\partial}{\partial s} \left(\alpha \frac{\partial v}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left(\beta \frac{\partial^2 v}{\partial s^2} \right) - \frac{\partial E_{ext}}{\partial s} = 0 \quad (\text{II.9})$$

Pour trouver une solution, le contour est rendu dynamique, en traitant $v(s)$ comme une équation du temps $v(s, t)$, représentant la position. La dérivée partielle par rapport au temps est posée égale à l'équation précédente [13] :

$$\begin{cases} \tau \frac{\partial v}{\partial t} = \frac{\partial}{\partial s} \left(\alpha(s) \frac{\partial v}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left(\beta(s) \frac{\partial^2 v}{\partial s^2} \right) - \frac{\partial E_{ext}}{\partial s} \\ v(0, s) = v_0(s), \quad \frac{\partial v}{\partial s}(t, 0) = \dot{v}_0(0) \end{cases} \quad (\text{II.10})$$

A l'équilibre, le terme de gauche de l'équation (II.10), tend vers zéro puis s'annule. Donnant, ainsi, une solution à l'équation voulue. C'est cette méthode qui fera l'objet de notre travail.

➤ **Algorithme glouton** [8]

Il est appelé aussi, algorithme greedy. Il est proposé par *William* et *Shah* en 1992. Il construit une solution de façon incrémentale en choisissant, à chaque étape, la direction qui est localement la plus prometteuse, dans un voisinage choisi ou préalable. D'un point de vue algorithmique, l'énergie est recalculée en chaque point du voisinage du nœud traité (Fig. II.3).

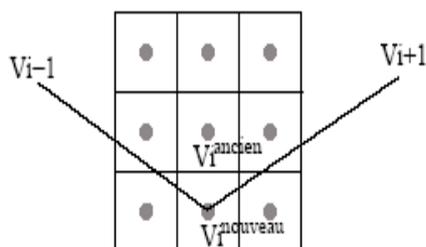


Fig. II.3 : le voisinage dans l'algorithme greedy.

Cependant, cet algorithme, par sa formulation (utilisation des fenêtres de calcul), est très sensible à la présence du bruit dans l'image.

II.3.3.2 Méthodes d'optimisation globale [8]

Il existe plusieurs méthodes d'optimisation globale. Ces méthodes permettent d'éviter le piègeage dans un minimum local. Parmi ces méthodes, on peut citer La programmation dynamique, le recuit simulé, les algorithmes génétiques, etc.

➤ **Programmation dynamique**

C'est une technique d'optimisation globale qui estime l'optimum d'une fonction de manière récursive. Son application aux contours actifs est due à *Amini* et *al.* en 1988. La convergence de la minimisation est garantie, mais la complexité est augmentée.

➤ **Le recuit simulé**

Le recuit simulé, est l'une des plus anciennes métaheuristiques qui dérive de la méthode de descente du gradient. Du fait de sa nature stochastique, la méthode peut s'avérer

très laborieuse en temps de convergence à condition que la solution soit prête de la solution voulue. Il fut introduit pour les contours actifs par *Berger* en 1991.

➤ **Algorithmes génétiques**

Un algorithme génétique est une procédure d'optimisation globale dont le principe s'inspire des mécanismes d'évolution des organismes vivants. Cet algorithme procède de manière itérative en faisant évoluer en parallèle une population de N individus, de génération en génération. Ils étaient appliqués aux contours actifs dans les travaux de *Rousselle* en 2003.

II.4 Implémentation des *snakes* par méthode variationnelle

Les méthodes variationnelles sont des méthodes itératives. À chaque itération n correspond une courbe C^n , définie par un ensemble de vecteurs positions :

$$v^n(s) = \begin{bmatrix} x^n(s) \\ y^n(s) \end{bmatrix} \quad (\text{II.11})$$

Avec :

$$\lim_{n \rightarrow \infty} v^n(s) = v^f(s) \quad (\text{II.12})$$

$x^n(s)$ et $y^n(s)$ sont les coordonnées d'un point de C^n dans le plan image.

On considère dans ce qui suit que : $\alpha(s) = \alpha = \text{cste}$, $\beta(s) = \beta = \text{cste}$ et $\lambda(s) \approx 1$. En

utilisant les coordonnées cartésiennes $x(s)$ et $y(s)$ telles que $v(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$, on aura :

$$\dot{x} = \frac{dx}{ds} \quad \dot{y} = \frac{dy}{ds} \quad \dot{v} = \frac{dv}{ds} \quad (\text{II.13})$$

$$\ddot{x} = \frac{d^2x}{ds^2} \quad \ddot{y} = \frac{d^2y}{ds^2} \quad \ddot{v} = \frac{d^2v}{ds^2} \quad (\text{II.14})$$

L'équation (II.10) s'écrit, alors :

$$\begin{cases} \tau \frac{\partial x}{\partial t} = \alpha \dot{x} - \beta x^{(4)} - \frac{\partial E_{ext}}{\partial x} \\ \tau \frac{\partial y}{\partial t} = \alpha \dot{y} - \beta y^{(4)} - \frac{\partial E_{ext}}{\partial y} \end{cases} \quad (\text{II.15})$$

Ainsi :

$$\|\dot{v}\|^2 = (\dot{x})^2 + (\dot{y})^2 \quad (\text{II.16})$$

$$\|\ddot{v}\|^2 = (\ddot{x})^2 + (\ddot{y})^2 \quad (\text{II.17})$$

L'énergie interne d'un point du contour actif s'écrit donc :

$$E_{int}(v(s)) = \frac{1}{2}(\alpha \|\dot{v}\|^2 + \beta \|\ddot{v}\|^2) \quad (\text{II.18})$$

Des équations (II.16) et (II.17) on obtient :

$$E_{int}(v(s)) = \frac{1}{2}(\alpha[(\dot{x})^2 + (\dot{y})^2] + \beta[(\ddot{x})^2 + (\ddot{y})^2]) \quad (\text{II.19})$$

Par définition le gradient de l'énergie s'écrit :

$$\nabla E_{snake} = \begin{pmatrix} \frac{\partial E_{snake}}{\partial x} \\ \frac{\partial E_{snake}}{\partial y} \end{pmatrix} \quad (\text{II.20})$$

En introduisant les équations d'Euler $\nabla E_{snake} = 0$, pour le calcul du gradient, on obtient :

$$\begin{cases} \frac{\partial E_{snake}}{\partial x} = \frac{\partial E_{ext}}{\partial x} - \frac{d}{ds} \left[\frac{\partial E_{int}}{\partial \dot{x}} \right] + \frac{d^2}{ds^2} \left[\frac{\partial E_{int}}{\partial \ddot{x}} \right] \\ \frac{\partial E_{snake}}{\partial y} = \frac{\partial E_{ext}}{\partial y} - \frac{d}{ds} \left[\frac{\partial E_{int}}{\partial \dot{y}} \right] + \frac{d^2}{ds^2} \left[\frac{\partial E_{int}}{\partial \ddot{y}} \right] \end{cases} \quad (\text{II.21})$$

D'autre part, de l'équation (II.19), on a :

$$\frac{\partial E_{int}}{\partial x} = \alpha \dot{x} \Rightarrow \frac{d}{ds} \left[\frac{\partial E_{int}}{\partial \dot{x}} \right] = \alpha \ddot{x} \quad (\text{II.22})$$

$$\frac{\partial E_{int}}{\partial \ddot{x}} = \beta \ddot{x} \Rightarrow \frac{d^2}{ds^2} \left[\frac{\partial E_{int}}{\partial \ddot{x}} \right] = \beta x^{(4)} \quad (\text{II.23})$$

On fait le même calcul pour la dérivée par rapport à y .

On remplaçant, dans l'équation (II.21), les termes de dérivées par rapport à x et y , on aura :

$$\begin{cases} \frac{\partial E_{snake}}{\partial x} = \frac{\partial E_{ext}}{\partial x} - \alpha \ddot{x} + \beta x^{(4)} \\ \frac{\partial E_{snake}}{\partial y} = \frac{\partial E_{ext}}{\partial y} - \alpha \ddot{y} + \beta y^{(4)} \end{cases} \quad (\text{II.24})$$

De l'équation (II.15) et de l'équation (II.24), on tire :

$$\begin{cases} \tau \frac{\partial x}{\partial t} = - \frac{\partial E_{snake}}{\partial x} \\ \tau \frac{\partial y}{\partial t} = - \frac{\partial E_{snake}}{\partial y} \end{cases} \quad (\text{II.25})$$

Dans ce qui suit, on considère le contour actif comme un ensemble de p points. On choisit une courbe fermée, c'est-à-dire : $V(0) = V(p - 1)$ En utilisant les *différences finies*, les dérivées sont calculées de façon approchée, tel que :

$$\dot{x} = \frac{x_{i+1} - x_i}{h} \simeq x_{i+1} - x_i \quad (\text{II.27.a})$$

$$\ddot{x} = \frac{x_{i-1} - 2x_i + x_{i+1}}{h^2} \simeq x_{i-1} - 2x_i + x_{i+1} \quad (\text{II.27.b})$$

$$x^{(4)} = \frac{x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} + x_{i+2}}{h^4} \simeq x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} + x_{i+2} \quad (\text{II.27.c})$$

Où h est le pas entre deux points (on suppose $h=1$, un pas d'un pixel) et $i = 0, \dots, p-1$.

$$\text{Posant :} \quad \begin{cases} f_x(i) = \left(\frac{\partial E_{ext}}{\partial x} \right)_{x_i} \\ f_y(i) = \left(\frac{\partial E_{ext}}{\partial y} \right)_{y_i} \end{cases} \quad (\text{II.28})$$

Après la discrétisation, les équations d'Euler deviennent :

$$\begin{cases} \left(\frac{\partial E_{snake}}{\partial x} \right)_{x_i} = \frac{\partial E_{ext}}{\partial x} + \alpha(-x_{i-1} + 2x_i - x_{i+1}) + \beta(x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} - x_{i+2}) \\ \left(\frac{\partial E_{snake}}{\partial y} \right)_{y_i} = \frac{\partial E_{ext}}{\partial y} + \alpha(-y_{i-1} + 2y_i - y_{i+1}) + \beta(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} - y_{i+2}) \end{cases} \quad (\text{II.29})$$

En développant on aura :

$$\begin{cases} \left(\frac{\partial E_{snake}}{\partial x} \right)_{x_i} = f_x(i) + \beta x_{i-2} + (-\alpha - 4\beta)x_{i-1} + (2\alpha + 6\beta)x_i + (-\alpha - 4\beta)x_{i+1} + \beta x_{i+2} \\ \left(\frac{\partial E_{snake}}{\partial y} \right)_{y_i} = f_y(i) + \beta y_{i-2} + (-\alpha - 4\beta)y_{i-1} + (2\alpha + 6\beta)y_i + (-\alpha - 4\beta)y_{i+1} + \beta y_{i+2} \end{cases} \quad (\text{II.30})$$

Ces dernières peuvent être mises sous forme matricielle suivante :

$$\begin{cases} AX + f_x(x, y) = \frac{\partial E_{snake}}{\partial x} \\ AY + f_y(x, y) = \frac{\partial E_{snake}}{\partial y} \end{cases} \quad (\text{II.31})$$

Avec :

$$A = \begin{bmatrix} 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta & -\alpha - 4\beta \\ -\alpha - 4\beta & 2\alpha + 6\beta & -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta \\ \dots & \dots \\ \dots & \dots \\ -\alpha - 4\beta & \beta & 0 & \dots & 0 & \beta & -\alpha - 4\beta & 2\alpha + 6\beta \end{bmatrix} \quad (\text{II.32})$$

X est le vecteur composé des abscisses des points du contour actif et Y le vecteur composé des ordonnées :

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{p-1} \end{bmatrix}, \quad f_x(x, y) = \begin{bmatrix} f_x(0) \\ \vdots \\ \vdots \\ f_x(p-1) \end{bmatrix}, \quad Y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{p-1} \end{bmatrix}, \quad f_y(x, y) = \begin{bmatrix} f_y(0) \\ \vdots \\ \vdots \\ f_y(p-1) \end{bmatrix} \quad (\text{II.33})$$

En écrivant l'équation (II.25) sous la forme discrète, on aura :

$$\begin{cases} \tau \frac{X^{i+1} - X^i}{h} = -\frac{\partial E_{snake}}{\partial x} \\ \tau \frac{Y^{i+1} - Y^i}{h} = -\frac{\partial E_{snake}}{\partial y} \end{cases} \Rightarrow \begin{cases} X^{i+1} = X^i - \frac{h}{\tau} \frac{\partial E_{snake}}{\partial x} \\ Y^{i+1} = Y^i - \frac{h}{\tau} \frac{\partial E_{snake}}{\partial y} \end{cases} \quad (\text{II.34})$$

On posant $\frac{h}{\tau} = \frac{1}{\gamma}$, on aura :

$$\begin{cases} \frac{\partial E_{snake}}{\partial x} = -\gamma(X^{i+1} - X^i) \\ \frac{\partial E_{snake}}{\partial y} = -\gamma(Y^{i+1} - Y^i) \end{cases} \quad (\text{II.35})$$

En remplaçant les coordonnées du gradient de l'énergie dans l'équation (II.31), on obtient :

$$\begin{cases} AX^{i+1} + \lambda f_x(X^{i+1}, Y^{i+1}) = -\gamma(X^{i+1} - X^i) \\ AY^{i+1} + \lambda f_y(X^{i+1}, Y^{i+1}) = -\gamma(Y^{i+1} - Y^i) \end{cases} \quad (\text{II.36})$$

Avec λ le coefficient de pondération introduit afin de compenser des valeurs élevées de l'énergie externe.

Supposant que les valeurs de f_x et f_y varient peu, entre deux itérations. Ce qui permet de faire les approximations suivantes:

$$\begin{cases} f_x(X^{i+1}, Y^{i+1}) \simeq f_x(X^i, Y^i) \\ f_y(X^{i+1}, Y^{i+1}) \simeq f_y(X^i, Y^i) \end{cases} \quad (\text{II.37})$$

L'équation (II.36) devient :

$$\begin{cases} AX^{i+1} + \lambda f_x(X^i, Y^i) = -\gamma(X^{i+1} - X^i) \\ AY^{i+1} + \lambda f_y(X^i, Y^i) = -\gamma(Y^{i+1} - Y^i) \end{cases} \quad (\text{II.38})$$

D'où :

$$\begin{cases} (A + \gamma I_d)X^{i+1} = \gamma X^i - \lambda f_x(X^i, Y^i) \\ (A + \gamma I_d)Y^{i+1} = \gamma Y^i - \lambda f_y(X^i, Y^i) \end{cases} \quad (\text{II.39})$$

Avec I_d la matrice identité. Après inversion de la matrice $(A + \gamma I_d)$, on aura une représentation matricielle déterminant la position des pixels du *snake* discrétisé à implémenter :

$$\begin{cases} X^{i+1} = (A + \gamma I_d)^{-1} (\gamma X^i - \lambda f_x(X^i, Y^i)) \\ Y^{i+1} = (A + \gamma I_d)^{-1} (\gamma Y^i - \lambda f_y(X^i, Y^i)) \end{cases} \quad (\text{II.40})$$

A partir de la formulation (II-40), la prise en compte d'énergie externe, autre que le simple gradient utilisé, se fait en remplaçant, simplement, le terme $f(X^i, Y^i)$ par sa nouvelle expression.

II.5 Exemple

Un exemple de convergence des contours actifs, sur une image test, est présenté sur la figure (II.4) [12]. Le contour actif est initialisé, manuellement, à proximité du contour réel de l'objet à segmenter.

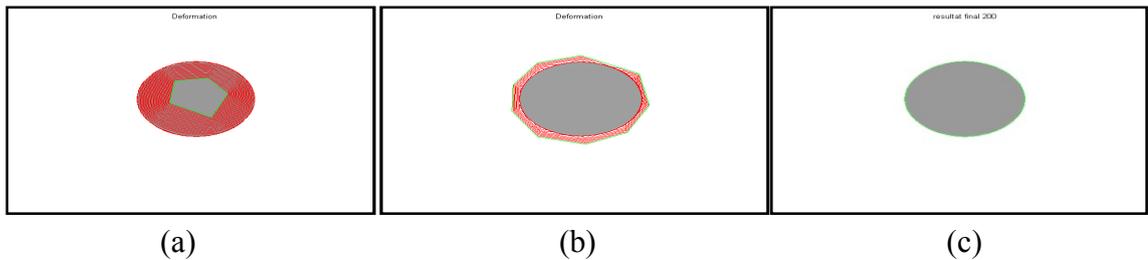


Fig. II.4 : exemple d'un contour actif : (a) initialisation à l'intérieur et évolution, (b) initialisation à l'extérieur et évolution et (c) résultat final.

Si on observe l'évolution de la courbe dans le temps, on constate que le contour actif commence par se contracter sous l'effet de l'énergie interne qui empêche les irrégularités : lissage d'abord, puis rigidité qui ramène les points du contour vers la frontière réelle. Le contour se fixe, enfin, sur les frontières grâce à l'énergie d'attache aux données.

II.6 Intérêts et limites des *snakes*

Les contours actifs, que nous venons d'introduire à travers leur formulation classique, permettent de segmenter efficacement des images de différentes natures sans connaissance *a priori* sur la géométrie des objets à extraire.

De plus, ce modèle offre à l'utilisateur la possibilité d'ajouter, à tout instant au cours de l'évolution du contour, des potentiels d'attraction ou de répulsion, plus au moins élaborés, qui vont permettre de contrôler sa convergence en temps réel.

Pour résumer, on peut dire qu'un contour actif a simplement besoin d'être initialisé, puisqu'il est autonome dans sa recherche de l'état d'énergie minimale. Cette propriété est très utile, notamment dans des applications particulières comme le suivi d'objet en mouvement dans les séquences d'images.

Malgré tous les avantages qu'il propose, le modèle des contours actifs soulève un certain nombre de limites :

- 1- le choix crucial de la position initiale : la courbe initiale doit être à proximité du contour souhaité.
- 2- la minimisation d'énergie demande l'inversion d'une matrice de grande taille, à chaque itération, impliquant un calcul très long.
- 3- le choix des paramètres, qui s'avère délicat, d'une image à une autre.
- 4- le changement de topologie de la courbe n'est pas vérifié. Ce qui implique une détection d'un seul objet, dans une image.
- 5- La présence de minima d'énergie locaux : qui peuvent amener le *snake* à converger vers une donnée inattendue.

Des nombreuses solutions ont été apportées aux problèmes cités. Concernant l'initialisation du *snake*, la solution la plus simple consiste à étendre la portée du champ gradient en épaississant les frontières par un filtrage gaussien, G_σ , d'écart type σ . Un des effets pervers de cette méthode est la délocalisation des frontières, qui induit des résultats moins précis. *Xu et Prince* [14] proposent, dans leur article, un nouveau champ de force : le flux de vecteur gradient (ou Gradient Vector Flow -GVF-, en anglais), qui permet à la fois de conserver les propriétés du gradient près des frontières, et de diffuser cette information dans des régions homogènes de l'image. C'est cette méthode qui fera l'objet de notre travail.

Pour les minimas locaux, qui ont tendance à retenir le contour actif dans son évolution, *Cohen* propose une amélioration significative grâce à l'introduction d'une nouvelle contrainte : la force ballon.

Pour le changement de topologie (plusieurs objets ne peuvent être détectés simultanément dans l'image) les recherches sont orientés vers la théorie de l'évolution des courbes et non plus vers la minimisation d'une fonctionnelle, débouchant sur le modèle des contours actifs géométriques.

II.7 La force ballon

Les contours actifs sont, généralement, basés sur l'information contour contenue dans l'image. Cependant, les forces issues des contours sont locales. Elles nécessitent l'initialisation du modèle près de la solution. En effet, au niveau des régions homogènes, là où il n'y a pas de contours, l'intensité des forces est proche de zéro. Ainsi, comme le montre la figure (II.5.(a)), le modèle ne peut être attiré vers les contours dans une zone homogène. Il est donc nécessaire d'avoir des forces qui étendent le modèle à l'intérieur de l'objet, y compris dans des zones homogènes avec une capacité à rétracter ce modèle s'il est en dehors (Fig. II.4.(b)).

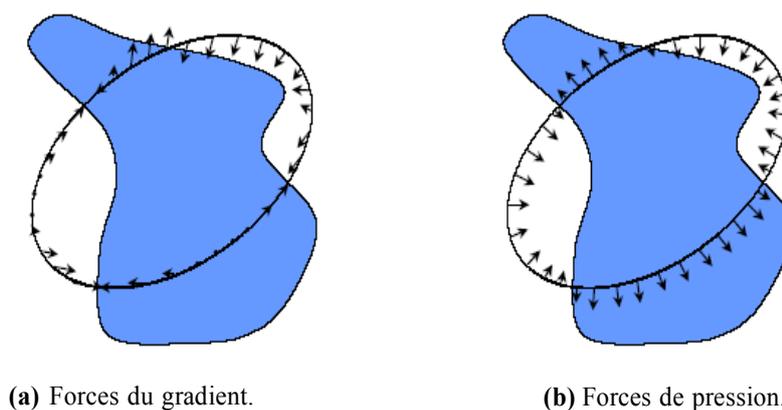


Fig. II.9 : forces issues du gradient de l'image et des forces de pression.

Afin de faciliter l'étape d'initialisation et pour conférer au *snake* des capacités de gonflement, *Cohen* a proposé une force de pression dite force ballon. Cette force, donnée par l'expression (II.42) [12], s'applique en chaque point du *snake*, en le poussant vers l'extérieur ou vers l'intérieur :

$$F_{ballon}(v(s)) = k_1 \vec{n}(v(s)) - k_2 |\nabla I(v(s))|^2 \quad (\text{II.42})$$

où $\vec{n}(v(s))$ représente le vecteur unité normal à la courbe en tout point $v(s)$ du *snake*, k_1 l'amplitude de la force (son signe donne le sens de la progression du *snake*). A l'équilibre, les forces ballon doivent contrebalancer les forces liées à l'image [12]. Pour ce faire, k_1 est pris légèrement plus petit que k_2 afin qu'un point contour puisse arrêter l'évolution du *snake*. Pour un coefficient k_1 de signe opposé qui signifie contraction, cette force accélère la constriction du *snake* sur lui-même l'empêchant ainsi d'être piégé par des zones de gradients parasites. La normalisation des forces extérieures implique que tout pixel de l'image influe de la même façon sur l'évolution des points du *snake*. En pratique, cette force est efficace dans le cas de formes circulaires ou sphériques et à niveaux de gradient quasi constant le long des contours. Toutefois, dès que les contours à atteindre deviennent complexes, la force ballon perd de son intérêt.

II.8 Flux du vecteur gradient

La force potentielle de distance est basée sur le principe que chaque point du *snake* devrait être attiré par les points appartenant au contour le plus proche. Ainsi, dans le cas de frontières présentant des concavités, les points du *snake* se retrouvent piégés par les point des murs de la concavité son jamais l'atteindre. Pour résoudre ce problème, Xu et Prince [15] ont proposé l'utilisation d'une équation de diffusion des vecteurs gradients d'une carte de contours dans des régions distantes des frontières. Cette équation permet d'agrandir le rayon de capture du *snake*. Ainsi, l'information contour est diffusée dans les régions homogènes, ce qui permet une initialisation éloignée du contour. Ceci donne lieu à un nouveau champ de force appelé Flux de Vecteur Gradient (*Gradient Vectors Flow*) qui est un champ vectoriel $w(x, y) = (u(x, y), v(x, y))$ qui minimise la fonctionnelle d'énergie suivante :

$$\varepsilon = \iint \mu (u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |w - \nabla f|^2 dx dy \quad (\text{II.46})$$

Où $f = f(x, y)$ représente une carte de contours de l'image que nous prendrons égale à l'énergie externe décrite dans la section (II.3.2). On peut voir par cette formulation variationnelle que quand les contours sont faible ($|\nabla f|$ et petit) alors l'énergie est dominée par la somme des dérivées partielles du champ vectorielle conduisant à un champ variant lentement. Au contraire quand le $|\nabla f|$ est grand, le second terme de l'intégrale domine et est minimisé pour une énergie égale au gradient du contour :

$$W = \nabla f \quad (\text{II.47})$$

Ainsi, le W est équivalent au gradient de la carte de contour quand le gradient du contour est important, ce qui revient à l'énergie externe traditionnelle, et varie peu dans les régions homogènes. μ , est un paramètre de régularisation commandant la différence entre les deux termes de l'intégrale. Ce paramètre se règle en fonction du bruit de l'image, plus le bruit est important plus ce paramètre doit être grand [14].

Le nouveau champ de force extérieure, W , remplace l'énergie externe de l'équation (II-10) :

$$\tau \frac{\partial v}{\partial t} = \frac{\partial}{\partial s} \left(\alpha(s) \frac{\partial v}{\partial s} \right) - \frac{\partial^2}{\partial s^2} \left(\beta(s) \frac{\partial^2 v}{\partial s^2} \right) + W \quad (\text{II.48})$$

La méthode GVF procède en deux étapes pour le calcul du champ GVF :

- le calcul de la carte de contours
- le calcul du champ GVF.

II.8.1 Calcul de la carte de contours

La carte de contours est dérivée de l'image. Elle permet de caractériser les zones d'intérêt dans l'image, c'est-à-dire, les contours dans notre cas, par les grandes valeurs, $f(x, y) = |\nabla G_\sigma * I(x, y)|$. Il vient alors que ∇f pointe sur les régions d'intérêt.

II.8.2 Le calcul du champ GVF

Le champ GVF est déterminé par le calcul variationnel. On obtient le système d'équations d'Euler suivantes à résoudre :

$$\begin{cases} \mu \nabla^2 u - (u - \nabla_x f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) = 0 \\ \mu \nabla^2 v - (v - \nabla_y f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) = 0 \end{cases} \quad (\text{II.49})$$

Où ∇_x et ∇_y sont respectivement les opérateurs gradient selon x et y . Ce système est résolu en traitant u et v comme des fonctions du temps et en le résolvant, de manière itérative de la même manière que pour la minimisation de l'énergie du modèle déformable (la méthode variationnelle).

On obtient alors :

$$\begin{cases} \frac{\partial u(x,y,t)}{\partial t} = \mu \nabla^2 u(x,y,t) - (u(x,y,t) - \nabla_x f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) \\ \frac{\partial v(x,y,t)}{\partial t} = \mu \nabla^2 v(x,y,t) - (v(x,y,t) - \nabla_y f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) \end{cases} \quad (\text{II.50})$$

Ces équations peuvent être résolues séparément en u et v . Elles sont connues sous le nom d'équations de diffusion généralisées.

On discrétisant en temps et en espace u et v on obtient :

$$\begin{cases} u(x,y,t+1) = u(x,y,t) + \Delta t \left[\mu \nabla^2 u(x,y,t) - (u(x,y,t) - \nabla_x f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) \right] \\ v(x,y,t+1) = v(x,y,t) + \Delta t \left[\mu \nabla^2 v(x,y,t) - (v(x,y,t) - \nabla_y f) \left((\nabla_x f)^2 + (\nabla_y f)^2 \right) \right] \end{cases} \quad (\text{II.51})$$

Pour simplifier, on réécrit les équations sous la forme suivantes :

$$\begin{cases} u(x,y,t+1) = (1 - b\Delta t)u(x,y,t) + \Delta t \mu \nabla^2 u(x,y,t) + \Delta t c^1(x,y) \\ v(x,y,t+1) = (1 - b\Delta t)v(x,y,t) + \Delta t \mu \nabla^2 v(x,y,t) + \Delta t c^2(x,y) \end{cases} \quad (\text{II.52})$$

Avec :

$$\begin{cases} b(x,y) = (\nabla_x f)^2 + (\nabla_y f)^2 \\ c^1(x,y) = \nabla_x f b(x,y) \\ c^2(x,y) = \nabla_y f b(x,y) \end{cases} \quad (\text{II.53})$$

Le laplacien de u et v peut s'exprimer sous la forme suivante :

$$\begin{cases} \nabla^2 u = \frac{1}{\Delta x \Delta y} (u(x+1,y) + u(x,y+1) + u(x-1,y) + u(x,y-1) - 4u(x,y)) \\ \nabla^2 v = \frac{1}{\Delta x \Delta y} (v(x+1,y) + v(x,y+1) + v(x-1,y) + v(x,y-1) - 4v(x,y)) \end{cases} \quad (\text{II.54})$$

Pour garantir la convergence, le pas en temps doit vérifier [14] :

$$\Delta t \leq \frac{\Delta x \Delta y}{4\mu} \quad (\text{II.55})$$

On voit bien, sur cette condition, que lors qu'on applique un lissage important (c'est-à-dire, Choisir μ grand) le pas du temps diminue par conséquent ; on aura une convergence lente. Lorsqu'on travaille sur une image à faible résolution (c'est-à-dire, Δx et Δy importants) la convergence devienne plus rapide. [14]

II-9 Flux du vecteur gradient généralisé

II.9.1 Principe et définition [14]

La méthode du contour actif GVF est une méthode originale dans sa conception. La force GVF présente beaucoup de propriétés souhaitable comme force externe, cependant on peut toujours apporter des améliorations pour créer une force plus sophistiquée. *XU* et *al* ont fait l'hypothèse que le lissage excessif du champ GVF engendré par le coefficient μ , près des contours, peut ne pas permettre au *snake* de converger vers les pics fins et longs des contours. Pour remédier à ce problème, ces auteurs ont proposés d'introduire deux fonctions de poids variant spatialement à la place du coefficient μ et $|\nabla f|^2$ dans l'équation (II.49), ceci donne une formulation généralisé du champ GVF. Ainsi, le champ GGVF est défini par la solution, à l'équilibre, de l'équation suivante

$$\frac{\partial W}{\partial t} = g(|\nabla f|)\nabla^2 W - h(|\nabla f|)(W - \nabla f) \quad (\text{II.56})$$

La fonction $g(|\nabla f|)$ est définie comme un **terme de lissage**, pour permettre l'adoucissement des variations du champ de vecteur. La fonction $h(|\nabla f|)$ est appelée **terme de donnée**. Elle permet de rapprocher le champ de vecteur W vers le ∇f [14]

II.9.2 Choix des fonctions de pondérations

Les deux fonctions dépendent des variations du gradient de la carte de contours, puisqu'on veut que le champ de vecteur soit lisse dans des zones loin des contours et soit proche du ∇f près des contours, alors la fonction $g(|\nabla f|)$ doit être dominante loin des contours et la fonction $h(|\nabla f|)$ doit être dominante près des contours. Dans le cas des pics fins et longs, le champ GVF simple est lissé partout du fait que le coefficient est constant pour toutes les valeurs du ∇f et y compris la région entre les contours opposées des pics fins. Ce qui réduit la force nécessaire pour pousser le *snake* à l'intérieure de la cavité [14].

Pour résoudre le problème, les fonctions de pondération doivent être choisies de façon à ce que $g(|\nabla f|)$ devienne faible lorsque $h(|\nabla f|)$ devienne importante. Ce qui permettra d'avoir à proximité des forts gradients un lissage très faible. Le champ de vecteur deviendra pratiquement égal aux gradients de la carte de contours

Xu et *al* ont proposés les deux fonctions de pondération suivantes :

$$g(|\nabla f|) = e^{-(|\nabla f|/k)} \quad \text{et} \quad h(|\nabla f|) = 1 - g(|\nabla f|) \quad (\text{II.57})$$

Où k est un coefficient qui quantifie la différence entre le lissage du champ et la conformité du gradient. Ici le lissage décroît exponentiellement en fonction du gradient de la carte contour et augmente, ainsi, l'importance de W .

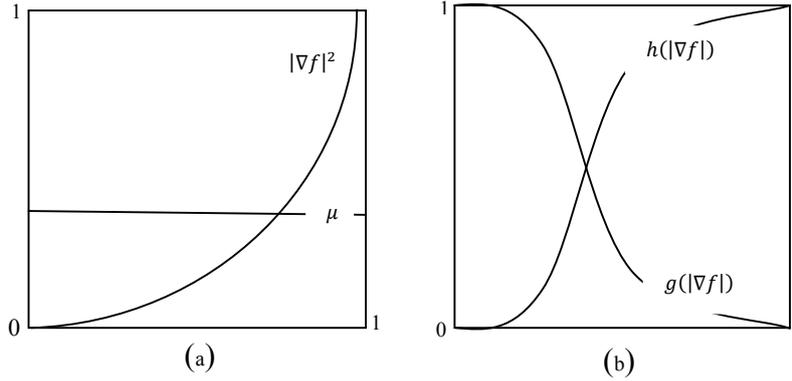


Fig. II.15 : illustration de la variation des fonctions de pondération : (a) GVF et (b) GGVF

II.9.3 Calcul du champ GGVF

Les composantes du champ GGVF sont calculées par un calcul variationnel en résolvant l'équation (II.54), qui s'écrit sous la forme développée en fonction de $u(x, y, t)$, et $v(x, y, t)$:

$$\begin{cases} \frac{\partial u}{\partial t} = g(|\nabla f|)\nabla^2 u - h(|\nabla f|)(u - \nabla_x f) \\ \frac{\partial v}{\partial t} = g(|\nabla f|)\nabla^2 v - h(|\nabla f|)(v - \nabla_y f) \end{cases} \quad (\text{II.58})$$

En discrétisant ces équations on aura :

$$\begin{cases} \frac{u(x, y, t+1) - u(x, y, t)}{\Delta t} = g(|\nabla f|)\nabla^2 u(x, y, t) - h(|\nabla f|)(u(x, y, t) - \nabla_x f) \\ \frac{v(x, y, t+1) - v(x, y, t)}{\Delta t} = g(|\nabla f|)\nabla^2 v(x, y, t) - h(|\nabla f|)(v(x, y, t) - \nabla_y f) \end{cases} \quad (\text{II.59})$$

D'où :

$$\begin{cases} u(x, y, t + 1) = u(x, y, t) + \Delta t [g(|\nabla f|)\nabla^2 u(x, y, t) - h(|\nabla f|)(u(x, y, t) - \nabla_x f)] \\ v(x, y, t + 1) = v(x, y, t) + \Delta t [g(|\nabla f|)\nabla^2 v(x, y, t) - h(|\nabla f|)(v(x, y, t) - \nabla_y f)] \end{cases} \quad (\text{II.60})$$

On pose :

$$\begin{cases} b(x, y) = h(|\nabla f|) \\ c^1(x, y) = \nabla_x f b(x, y) \\ c^2(x, y) = \nabla_y f b(x, y) \end{cases} \quad (\text{II.61})$$

En remplaçant dans l'équation (II.58) on obtient :

$$\begin{cases} u(x, y, t + 1) = (1 - b\Delta t)u(x, y, t) + \Delta t g(|\nabla f|)\nabla^2 u(x, y, t) + \Delta t c^1(x, y) \\ v(x, y, t + 1) = (1 - b\Delta t)v(x, y, t) + \Delta t g(|\nabla f|)\nabla^2 v(x, y, t) + \Delta t c^2(x, y) \end{cases} \quad (\text{II.62})$$

Pour garantir la convergence, le pas du temps doit satisfaire la condition suivante :

$$\Delta t \leq \frac{\Delta x \Delta y}{4g_{max}} \quad (\text{II.63})$$

Cependant, avec toutes les améliorations portées sur le *snake*, il reste le problème de changement de topologie.

II.10 contours actifs géométriques et géodésiques

Le modèle contours actifs géométriques découle d'une approche dite approche eulérienne, différente de celle des *snake*. Il est basé sur la théorie de l'évolution des courbes planes et des équations aux dérivées partielles (EDP) géométriques. Ces modèles représentent, implicitement, les courbes et les surfaces comme un ensemble de niveaux d'une fonction de plus haute dimension. Le contour est considéré comme une ligne de niveau zéro d'une surface.

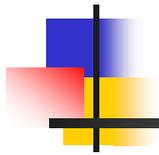
Le modèle des contours actifs géodésiques a été proposé par *Caselle et al*, après le contour actif géométrique. Il a « réconcilié » l'approche variationnelle et l'approche directe par évolution de courbes. Dans ce modèle, résoudre le problème du contour actif est équivalent à chercher un chemin de distance minimale, appelé courbe géodésique.

Ces nouvelles approches permettent le traitement de topologie, ce qui donne la caractéristique de pouvoir se diviser et donc, de détecter plusieurs objets dans l'image. Cependant, ces modèles augmentent la dimension du système ce qui induit des calculs colossaux et, ainsi, un temps d'exécution lent.

II.11 conclusion

Dans ce chapitre, nous avons étudié la théorie des contours actifs afin de mieux comprendre leur principe. Nous avons développé le modèle initial : *snake*. Mais nous nous sommes accentués sur le *snake GVF*. L'évolution du *snake GVF* est régie par les énergies internes et l'énergie du flux du vecteur du gradient qui représente l'énergie externe. Ces deux énergies dépendent des paramètres de régularisation, qu'il faudra choisir de sorte à assurer une bonne convergence du modèle. Cette méthode présente beaucoup d'avantages. Elle permet la détection des zones de fortes concavités et de détecter des frontières peu visibles. Il n'est pas nécessaire de savoir si le modèle doit croître ou se rétracter vers le contour souhaité, au contraire de la force ballon. Le *snake GVF* possède aussi une plage de capture plus importante permettant ainsi une initialisation éloigné du contour. Néanmoins, une initialisation proche des contours à détectés est toujours nécessaire, surtout lorsque ils' agit des images réel à plusieurs objets. Aussi, il ne peut pas changer sa topologie, contrairement aux contours actifs géométriques et géodésiques, donc il ne permet pas la détection de plusieurs objets.

La méthode du *snake GVF* est une méthode semi-automatique, elle nécessite une initialisation manuelle du contour initial. Dans le chapitre suivant nous allons implémenter le *snake GVF*, et nous allons proposer une solution d'automatisation de l'initialisation du modèle.



Chapitre III

Approche adoptée

III.1 Introduction

Dans le chapitre précédent, nous avons développé les modèles des contours actifs GVF et GGVF, qui représentent deux méthodes de détection de contours. Afin d'élaborer un processus de segmentation, et dans le souci d'établir une comparaison entre le *snake* classique, le *snake* GVF et le *snake* GGVF, nous allons programmer sous Matlab ces trois méthodes.

En premier lieu, nous allons exposer notre approche adoptée, en donnant des exemples sur des images tests. Nous nous penchons sur l'approche par le *snake* GVF et GGVF. A la fin, nous allons aborder l'initialisation automatique du *snake*.

III.2 Langage de programmation

Le nom MATLAB provient de la contraction de MATrix LABoratory. C'est un langage de calcul scientifique basé sur le type de variable matricielle. Ses grandes capacités de calcul numérique lui permettent d'être appliqué à différents domaines tels traitement du signal, automatisme, etc.

L'utilisation de MATLAB peut se faire de façon itérative en exécutant des commandes directement après le prompt ou par des scripts. Ces derniers peuvent être des simples programmes ou des fonctions avec des paramètres d'appel et de retour.

Matlab est aujourd'hui un langage de programmation complet dans un environnement de développement simple et pratique. Pour ce la, nous avons implémenté notre travail avec le langage MATLAB sur un PENTIUM IV (CPU 3.20 GHz 1Go de RAM).

III.3 Approche adoptée

L'idée de base est de faire évoluer le *snake* en minimisant son énergie, à fin d'épouser la forme du contour ou de l'objet qu'on veut détecter, et ceci tout en agissant sur un contour initial déterminé par l'opérateur. Pour l'énergie externe, en question, nous allons utiliser dans une première approche l'énergie GVF simple qui est donnée par :

$$E_{ext} = kV_{GVF} \quad (III.1)$$

Le modèle paramétrique global est obtenu en minimisant l'énergie suivante :

$$E_{snake} = \frac{1}{2} \int_0^1 \left[\alpha \left| \frac{dv}{ds} \right|^2 + \beta \left| \frac{d^2v}{ds^2} \right|^2 \right] ds - kV_{GVF} \quad (III.2)$$

Une deuxième approche consiste à remplacer dans l'équation (III.2) l'énergie externe par l'énergie GGVF donnée par :

$$E_{ext} = kV_{GGVF} \quad (III.3)$$

La résolution de ces deux approches est basée sur le calcul variationnelle, qui est une méthode simple à implémenter

III.4 Implémentation

Pour réaliser l'implémentation des deux algorithmes GVF et GGVF, nous avons organisé notre programme suivant l'organigramme de la figure suivante :

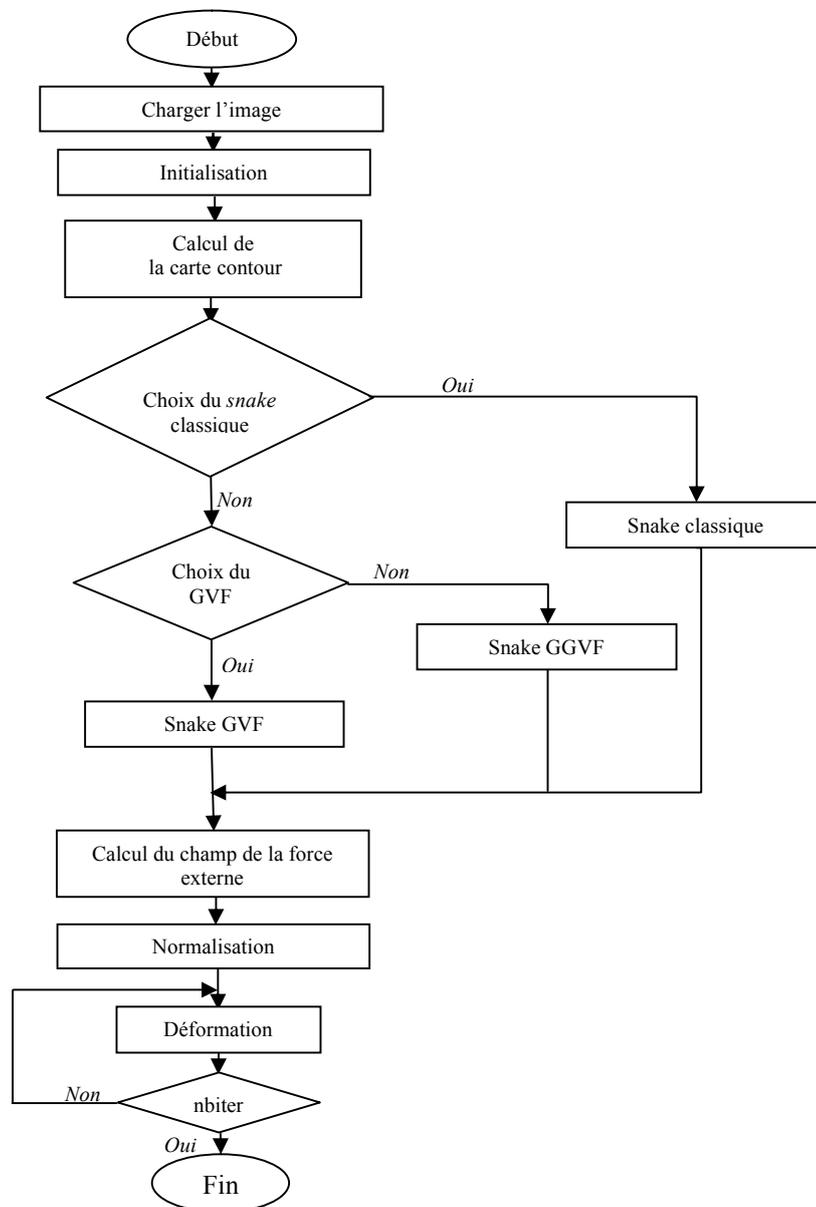


Fig. III.1 : organigramme de l'algorithme d'implémentation.

III.4.1 Lecture de l'image et Calcul de la carte contour

Cette partie de l'algorithme permet la lecture de la matrice des niveaux de gris de l'image, stockée dans le fichier dont le chemin est spécifié en argument, et retourne l'image sur une interface (Fig.III.2), qui permet à l'opérateur de choisir les points du contour initial par des simples clics sur les points à l'aide de la souris. Les points sélectionnés sont interpolés pour construire le contour initial.

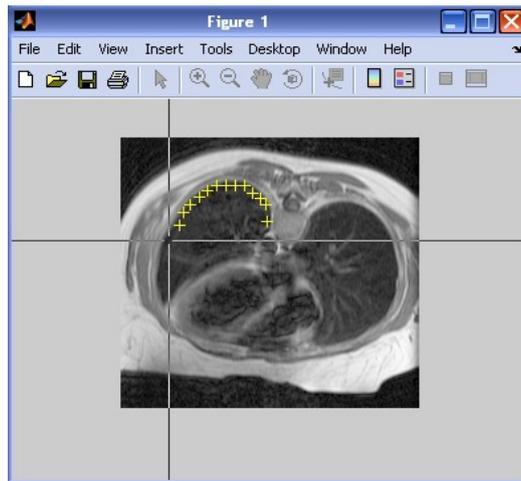


Fig. III.2 : illustration du choix des points du contour initial.

III.4.2 interpolation et reparamétrisation

Pendant l'évolution, les points du contour sont interpolés pour produire un *snake* continu. Pour l'interpolation nous avons choisit l'interpolation linéaire prenant deux points successifs et dessinant un segment de droite entre ces deux points.

Durant la déformation, deux points voisins, peuvent se rapprocher l'un de l'autre ou, au contraire s'éloigner. L'espacement entre les points varie, donc, d'une itération à l'autre. Des points seront ajoutés là où cet espacement devient trop important et seront enlevés quand l'espacement est trop faible. C'est la reparamétrisation, pour ce faire on définit deux valeurs de distance (d_{min} et d_{max}) et on calcule la distance entre deux points successifs par la formule suivante :

$$d = \sqrt{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)} \quad (\text{III.4})$$

Ainsi, Si $d < d_{min}$ alors, l'un de ces points est enlevé. Si $d > d_{max}$ alors, un point est ajouté entre ces deux points. Dans notre cas $(d_{min}, d_{max}) = (0.5, 2)$.

III.4.3 Choix du *snake* classique, GVF ou GGVF

Après la validation du choix du contour initial, l'algorithme retourne un menu (Fig III.3) à partir du quel l'opérateur va choisir le type de *snake* à appliquer (*snake* GVF, *snake* GGVF ou *snake* classique).



Fig. III.3 : menu du choix du GVF.

III.4.4 Calcul de la carte contour

En général, la carte contour $f(x, y)$ (Fig. III.4(b)) est calculée par la formule suivante :

$$f(x, y) = |\nabla G_{\sigma} * I(x, y)| \quad (\text{III.5})$$

Nous avons choisit d'accentuer les contours en utilisant l'image des contours détectée par le filtre de *canny*, à la place de $I(x, y)$ (Fig. III.4 (c)).

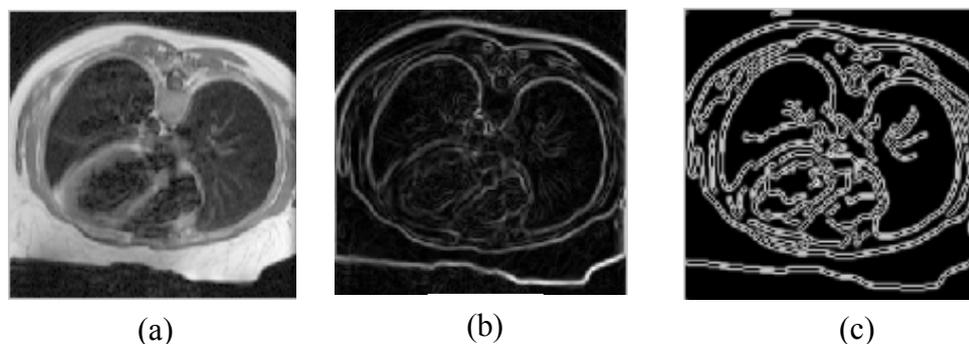


Fig. III.4 : exemple de carte contour : (a) image originale, (b) carte contour classique et (c) carte contour à base de l'image filtrée.

III.4.5 Calcul du champ GVF

Le champ GVF (Fig.III.5) est défini par le champ vectoriel $w(x, y) = (u(x, y), v(x, y))$ qui minimise la fonctionnelle d'énergie expliquée dans le chapitre précédant (équation II.46). Les composantes du champ GVF sont déterminées par un calcul variationnel, en résolvant le système d'équations d'Euler (équation II.49). Après discrétisation, nous aurons :

$$\begin{cases} u(x, y, t + 1) = (1 - b\Delta t)u(x, y, t) + \Delta t\mu\nabla^2 u(x, y, t) + \Delta tc^1(x, y) \\ v(x, y, t + 1) = (1 - b\Delta t)v(x, y, t) + \Delta t\mu\nabla^2 v(x, y, t) + \Delta tc^2(x, y) \end{cases} \quad (\text{III.6})$$

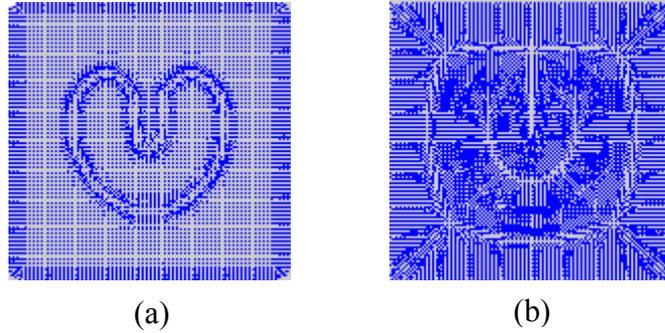


Fig. III.5 : exemple de calcul du champ GVF : (a) gradient de la carte contour et (b) carte du champ GVF.

III.4.6 Calcul du champ GGVF

Le champ GGVF (Fig. III.6) est défini par la solution, à l'équilibre, de l'équation (II.54). Sachant que $g(|\nabla f|)$ permet l'adoucissement des variations du champ de vecteur, et $h(|\nabla f|)$ permet de rapprocher le champ de vecteur W vers le ∇f . Après discrétisation les composantes du champ sont calculées comme suit :

$$\begin{cases} u(x, y, t + 1) = (1 - b\Delta t)u(x, y, t) + \Delta tg(|\nabla f|)\nabla^2 u(x, y, t) + \Delta tc^1(x, y) \\ v(x, y, t + 1) = (1 - b\Delta t)v(x, y, t) + \Delta tg(|\nabla f|)\nabla^2 v(x, y, t) + \Delta tc^2(x, y) \end{cases} \quad (\text{III.7})$$

Avec : $g(|\nabla f|) = e^{-(|\nabla f|/k)}$, $h(|\nabla f|) = 1 - g(|\nabla f|)$, $b = h(|\nabla f|)$, $c_1 = b \nabla_x f$ et $c_2 = b \nabla_y f$.

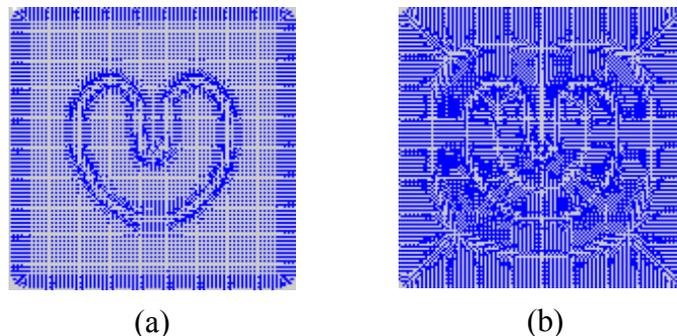


Fig. III.6 : exemple de calcul du champ GGVF : (a) gradient de la carte contour et (b) carte du champ GGVF.

III.4.7 Déformation du *snake*

La déformation (Fig. III.7) est assurée par un module, qui prend en argument les composantes du contour initial, le jeu des paramètres α , β , γ , λ et les composantes du champ GVF ou GGVF. En premier lieu, il calcule les éléments de la matrice de rigidité A (équation II.32) toute en calculant son inverse. Les composantes du prochain *snake* sont calculées par les équations suivantes :

$$\begin{cases} X^{i+1} = (A + \gamma I_d)^{-1} (\gamma X^i - \lambda f_x(X^i, Y^i)) \\ Y^{i+1} = (A + \gamma I_d)^{-1} (\gamma Y^i - \lambda f_y(X^i, Y^i)) \end{cases} \quad (\text{III.8})$$

En faisant une interpolation linéaire de ses points, nous obtenons le contour déformé.

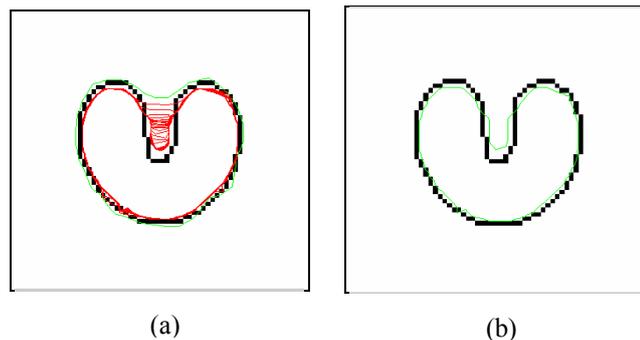


Fig. III.7 : (a) processus de déformation du *snake* GVF.
(b) résultat final.

III.5 Critère d'arrêt

Dans le cas idéal, le *snake* s'arrête quand le minimum d'énergie est atteint, c'est-à-dire, lorsque les points du *snake* ne bougent pas. Mais dans la pratique, ceci n'est pas envisageable. C'est à l'utilisateur de décider quand s'arrête le processus de déformation. Dans notre application, nous avons pris, comme critère d'arrêt, le nombre d'itérations qui sera initialisé par l'utilisateur.

Dans la mise en œuvre effective de l'algorithme, la décision de déplacement du *snake* est liée à chaque énergie. Les différents paramètres pris en compte influent aussi sur les contours.

III.6 Choix des paramètres de l'approche

Les différents paramètres du *snake* diffèrent d'une image à une autre. Ils sont déterminés, empiriquement, à travers une série de tests, en s'appuyant sur les caractéristiques de l'objet à détecter dans l'image.

a) Terme d'élasticité α

α représente la contrainte d'élasticité. Il exprime la faculté, pour chaque point, de s'écarter de ses voisins. Pour une valeur trop petite, les nœuds s'éloignent facilement les uns des autres, ce qui piège les nœuds proches des minimas locaux, et induit une déstabilisation du *snake*. Une valeur très grande rend les nœuds très solidaires entre eux et difficile à s'écarter. Ce qui favorise l'effondrement du *snake* sur lui même.

b) Terme de rigidité β

β est lié à la rigidité de la courbe. Ainsi, si on veut détecter précisément des angles, on aura tendance à diminuer ce paramètre. Au contraire, si l'on veut privilégier une certaine régularité, un β élevé arrondira les angles.

c) Le pas du temps γ

Le pas du temps est ajusté de telle sorte que le déplacement de chaque nœud ne dépasse pas un ou deux pixels. Le choix du pas du temps γ est, donc, déterminant, pour l'évolution de la courbe. Si γ est trop grand, la courbe risque de passer par-dessus le contour.

d) Le poids du champ GVF μ

En changeant les valeurs de μ dans l'équation (III.6), nous obtenons les différents résultats présentés dans la figure (III.8).

La valeur de μ conditionne l'équilibre de la force externe GVF. D'autant μ est grand, d'autant les régions d'intérêts (les contours) disparaissent. Cependant, si μ est faible il y a lieu d'éliminer la force GVF.

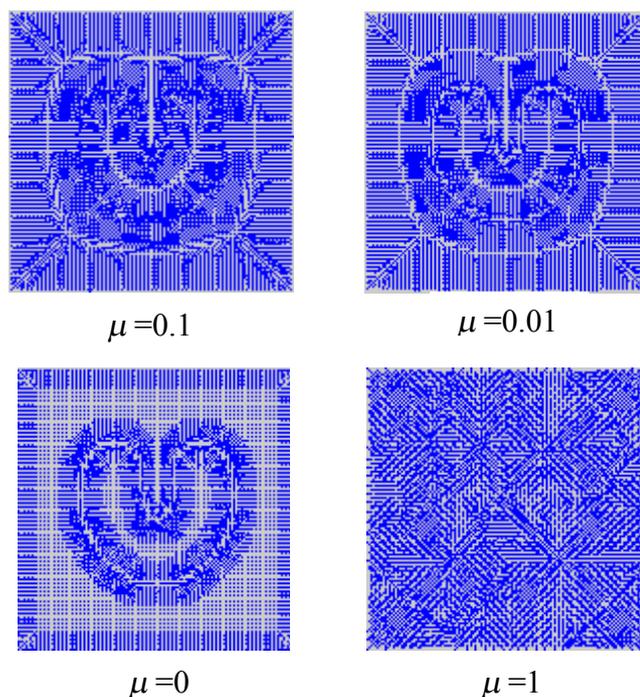


Fig. III.8 : les différents résultats du GVF en fonction du paramètre μ .

III.7 Initialisation automatique

Le *snake*, que nous avons adopté, travaille à partir d'un contour initial introduit, manuellement, par l'opérateur. Vue sa sensibilité à l'initialisation, l'intervention de l'opérateur peut réduire ses performances. Pour cela, nous avons pensé à améliorer l'algorithme (Fig. III.1), en introduisant une méthode d'initialisation automatique, en réduisant l'intervention de l'opérateur au minimum.

L'initialisation automatique du *snake* est réalisée par une analyse de l'histogramme et par différents traitements de bas niveaux. L'ensemble des étapes de recherche du contour initial permet d'obtenir un contour proche du contour réel.

III.7.1 Analyse de l'histogramme

En général, l'histogramme d'une image IRM est composé de trois classes de niveaux de gris (le fond, les organes mous et les os avec les vaisseaux sanguin). Ces classes peuvent être estimées par trois gaussiennes. L'histogramme peut être modélisé par une somme de densités des gaussiennes :

$$G(x) = \sum_{i=1}^{\alpha} w_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\left(\frac{-1}{2\sigma_i^2}(x^2 - m_i^2)\right)} \quad (\text{III.9})$$

σ_i , m_i , w_i représentent respectivement la variance, la moyenne et le poids de chacune des α gaussiennes ($\alpha = 3$).

Pour ajuster une gaussienne, nous estimons donc ses paramètres (σ_i , m_i , w_i). L'estimation est réalisée par un algorithme dit *algorithme de demelange* [18], qui utilise la méthode du maximum de la vraisemblance (Annexe A).

Un exemple d'image et son histogramme, estimé par l'algorithme de *demelange*, est représenté sur la figure (III.9).

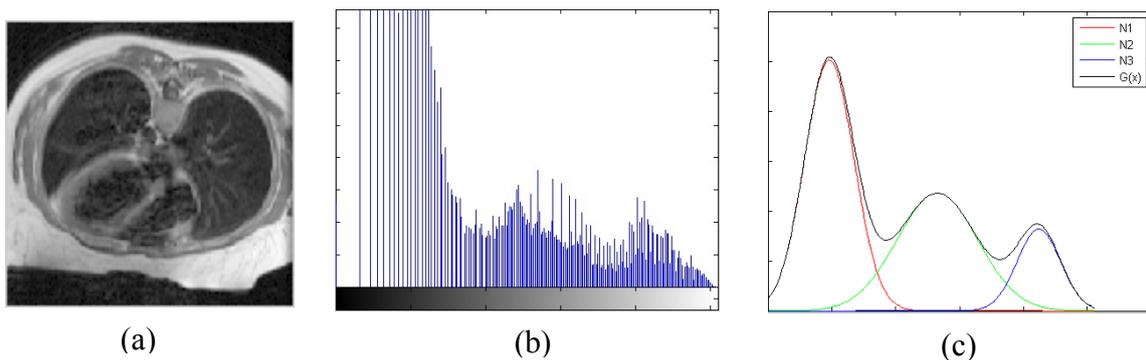


Fig. III.9 : exemple de modélisation d'historgramme : (a) image originale, (b) histogramme et (c) histogramme estimé.

Sur l'historgramme estimé, nous distinguons facilement les différentes classes des niveaux de gris. Ce qui nous permet de définir le seuil ou les seuils délimitant la région d'intérêt.

III.7.2 Sélection de la classe de la région d'intérêt

La région d'intérêt est sélectionnée en réalisant un seuillage à partir des seuils définis précédemment (Fig. III.10).

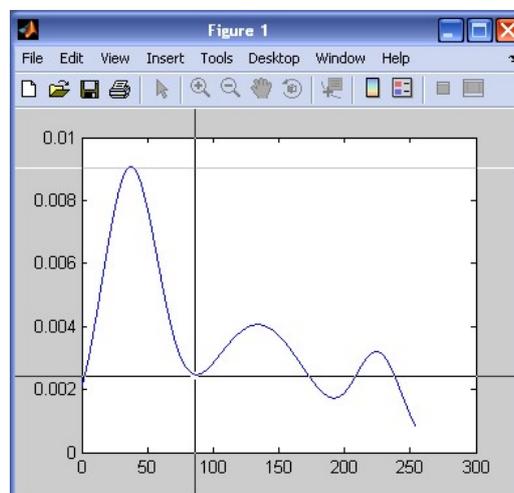


Fig. III.10 : choix des seuils.

Après le seuillage, nous obtiendrons une image binaire (Fig. III.11). La multiplication de l'image originale par l'image binaire, nous donne une image mettant en évidence la classe de la région d'intérêt (Fig.III.12).

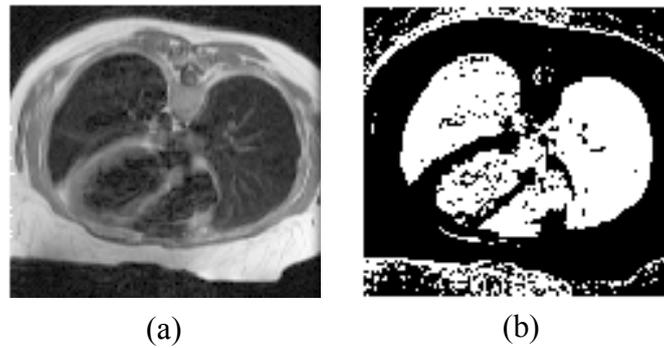


Fig. III.11 : le résultat après seuillage : (a) image original et (b) image binaire.

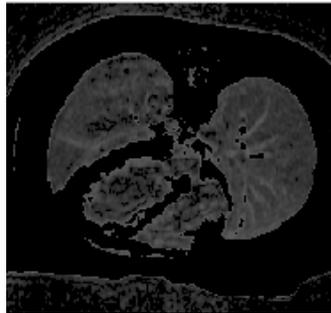


Fig. III.12 : résultat de la multiplication.

III.7.3 Détection de contour

Afin de différencier les organes, nous effectuons une détection de contour. Les points contours seront, alors, soustraits de l'image seuillée, afin de créer une vallée entre chaque organe. Nous appliquerons, tout d'abord, un filtrage médian sur l'image seuillée, avec une taille de fenêtre égalé à 5, pour enlever les points aberrants, sans dégrader les contours (Fig. III.13(a)). Nous réaliserons, ensuite, la détection des points de contour par un filtre optimal : le filtre de *Canny*. Le résultat de se filtrage permet de détecter les contours de la région d'intérêt (Fig.III.13 (b)).

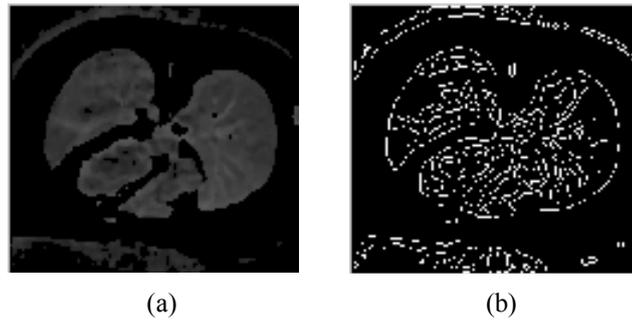


Fig. III.13 : exemple de détection de contour : (a) image filtrée et (b) image avec contours accentués.

III.7.4 Sélection de la région d'intérêt

Les points qualifiés de points contour, par le traitement précédent, sont alors soustraits de l'image binarisée. La soustraction permet de séparer les différents organes constituant la région d'intérêt (Fig. III.14).

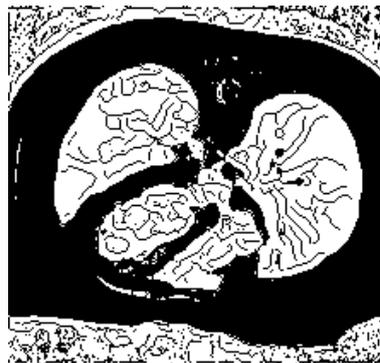


Fig. III.14 : soustraction des points de contours de l'image binarisée.

Nous mettons alors en évidence la région d'intérêt. Nous réaliserons, ensuite, un remplissage de trous grâce à un filtre morphologique (Fig. III.15).

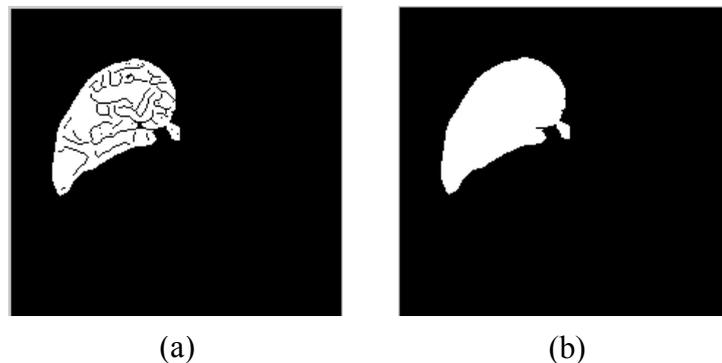


Fig. III.15 : (a) la région sélectionnée, (b) le remplissage des trous.

La zone d'intérêt, ainsi définie, est érodée pour s'assurer que le contour soit à l'intérieur de la région à détecter. Le contour de la zone érodée constitue, alors, le contour initial (Fig. III.16).

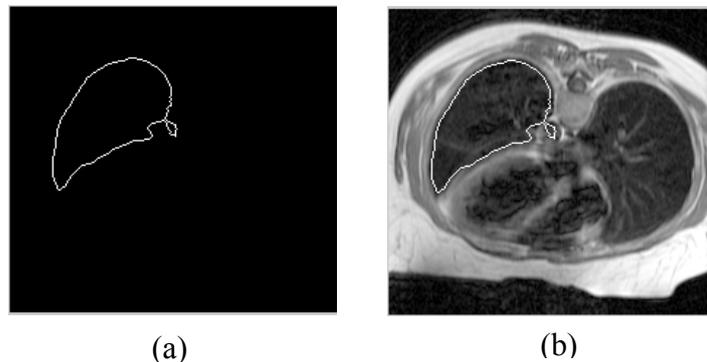


Fig. III.16 : (a) contour initial détecté, (b) superposition du contour initial sur l'image originale.

Nous constatons que le contour initial est très proche du contour réel et reproduit sa forme. Ainsi, ce processus d'initialisation est introduit à la place de l'étape d'initialisation manuelle de l'algorithme de la figure(III.1) (Fig.III.17).

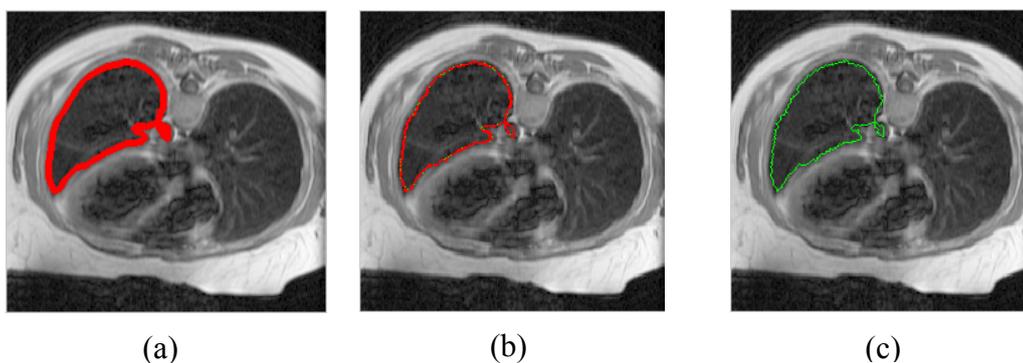


Fig. III.17 : exemple de déformation d'un *snake* initialisé automatiquement: (a) contour initial, (b) contour déformé et (c) contour final.

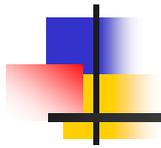
III.8 Conclusion

Dans ce chapitre nous avons expliqué l'implémentation des deux *snakes* GVF et GGVF, qui représentent deux modèles de détection de contour.

Ces deux modèles possèdent la faculté de détecter des formes complexes et irrégulières. Cependant, ils nécessitent une initialisation proche du contour à détecter. Pour

cela, nous avons proposé une méthode d'initialisation qui nous permet de choisir le contour initial automatiquement. Ainsi, nous avons réalisé un processus de segmentation automatique.

Afin de valider notre méthode, nous allons l'appliquer à différentes images IRM dans le chapitre suivant.



Chapitre IV

Tests et résultats

IV.1 Introduction

Dans ce chapitre nous nous intéressons au problème de détection de tumeur dans une IRM. L'objectif est de trouver les contours de la tumeur en appliquant à l'image l'approche adoptée. Cependant, afin de valider notre algorithme et surtout l'efficacité de celui-ci, nous avons, tout d'abord, entrepris différents tests sur une image test : image *chest*.

Ainsi, ce chapitre est organisé de la manière suivante. En premier lieu, nous allons présenter les résultats sur l'image test, en lui appliquant les trois types de *snakes* : à savoir le *snake* classique, le *snake* GVF et le *snake* GGVF, et ceci avec les deux méthodes d'initialisation, manuelle et automatique. Ensuite, nous allons les appliquer sur une image IRM cérébrale afin de détecter une tumeur.

IV.2 Application de l'approche sur l'image test

Pour effectuer nos tests, nous avons choisi l'image test *chest* (Fig. IV.1), qui est une image de dimension 211x225. Pour cette image nous avons testé plusieurs valeurs des paramètres : α (élasticité), β (rigidité), γ (pas du temps), μ (le poids du champ GVF). Après un certain nombre de tests, nous avons choisi : $\alpha = 0.05$; $\beta = 0$; $\gamma = 1$; $\mu = 0.15$.

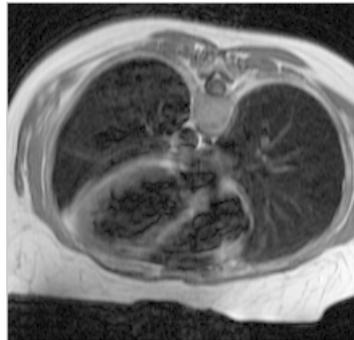


Fig. IV.1: l'image test chest.

IV.2.1 Initialisation manuelle

Nous allons commencer à appliquer une initialisation manuelle de nos trois *snakes*. Ensuite nous appliquerons l'initialisation automatique développée en chapitre précédent. Ainsi, les figures IV.2, IV.3 et IV.4, montrent le résultat de détection des contours, d'une région d'intérêt, par initialisation manuelle d'un *snake* classique, *snake* GVF et *snake* GGVF, respectivement.

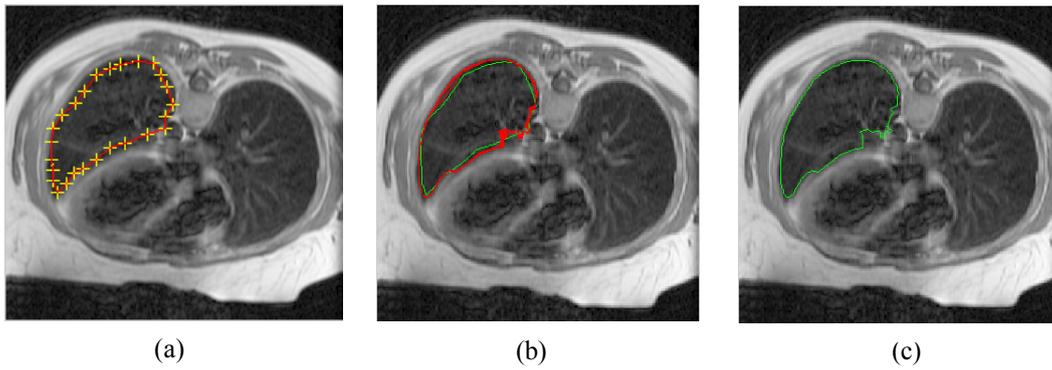


Fig. IV.2 : test du *snake* classique : (a) image originale avec initialisation manuelle, (b) déformations, (c) résultat après 80 itérations.

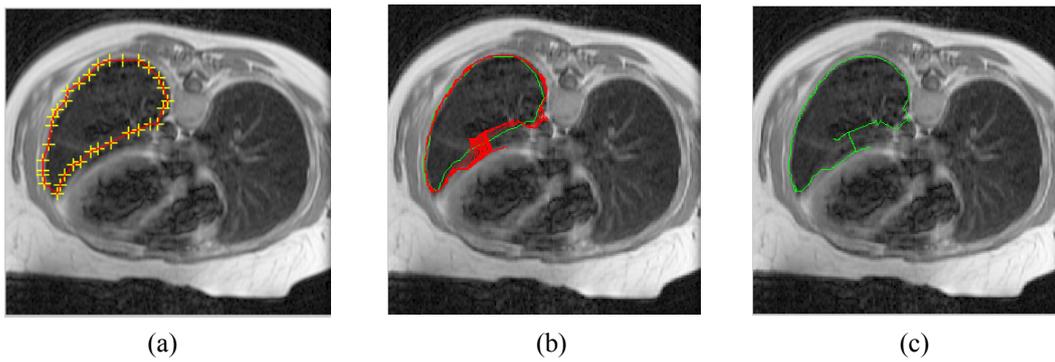


Fig. IV.3 : test du *snake* GVF classique. (a) image originale avec initialisation manuelle, (b) déformations, (c) résultat après 80 itérations.

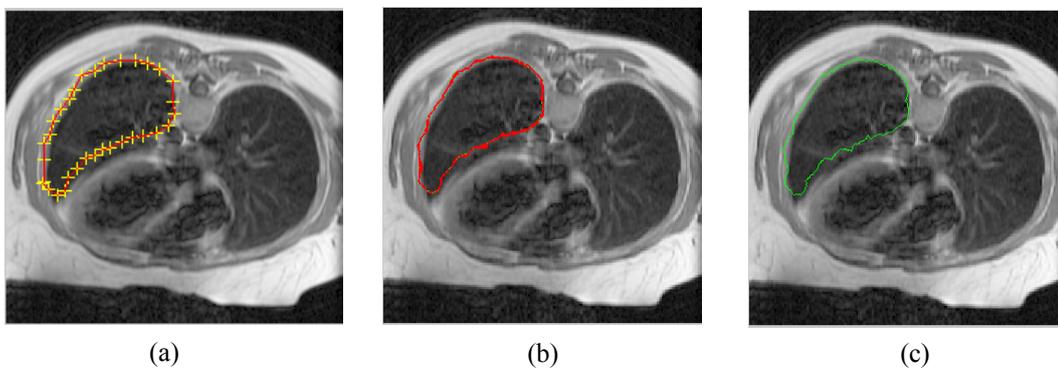


Figure IV.4 : test du *snake* GGVF : (a) image originale avec initialisation manuelle, (b) déformations, (c) résultat après 80 itérations.

IV.2.2 Initialisation automatique

De même, nous allons appliquer une initialisation automatique que nous avons développé à la section (III.7) du chapitre III, en l'appliquant sur les trois *snakes*. Ainsi, les figures IV.5, IV.6 et IV.7, montrent le résultat de détection des contours, d'une région d'intérêt, par initialisation automatique d'un *snake* classique, *snake* GVF et *snake* GGVF, respectivement.

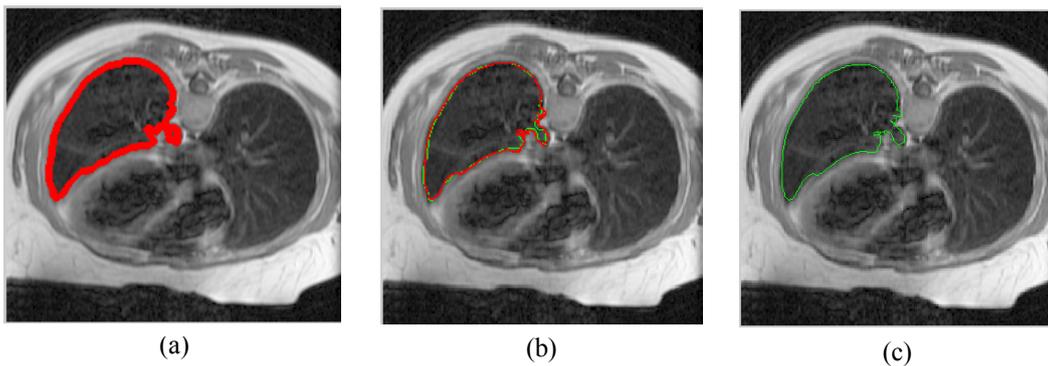


Fig. IV.5 : test du *snake* classique : (a) image originale avec initialisation automatique, (b) déformations, (c) résultat après 10 itérations.

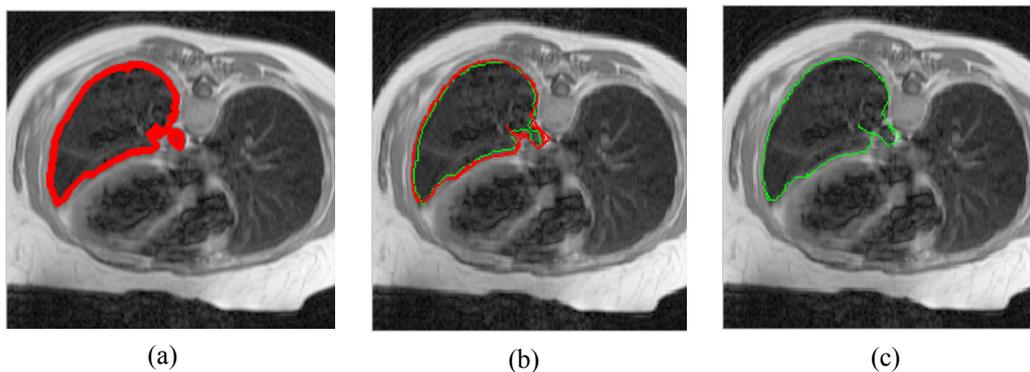


Fig. IV.6 : test du *snake* GVF : (a) image originale avec initialisation automatique, (b) déformations, (c) résultat après 10 itérations.

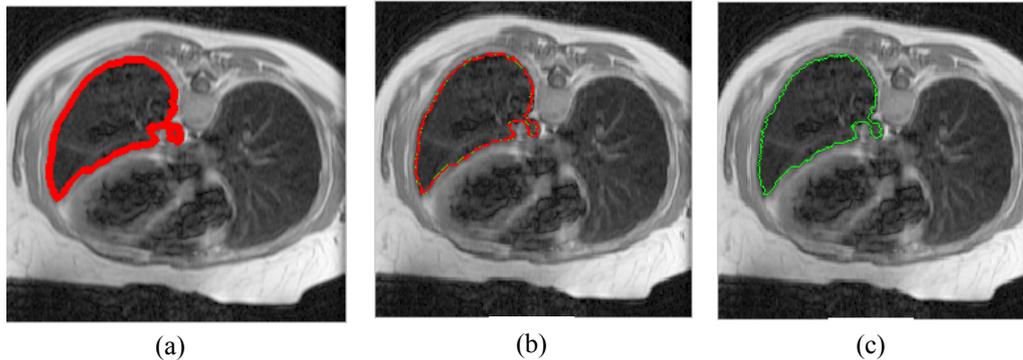


Fig. IV.7 : test du *snake* GGVF : (a) image originale avec initialisation automatique, (b) déformations, (c) résultat après 10 itérations.

IV.2.3 Interprétation des résultats

L'ensemble des résultats obtenus par initialisation automatique montre qu'il y a une nette amélioration de la détection des contours de la région d'intérêt. Ceci est dû au fait que le contour initialisé automatiquement est très proche du contour réel. En effet, le *snake* initial a presque la forme du contour réel. D'ailleurs, dès la première itération le *snake* détecte presque exactement le contour réel de la région d'intérêt. Ce qui diminue, ainsi, le nombre d'itérations qui est, considérablement, réduits de 80 à 10 itérations.

En comparant les résultats du point de vue type du *snake* utilisé, nous remarquons que pour le modèle du *snake* GVF classique avec initialisation manuelle, la détection est mauvaise (Fig. IV.3(c)). Sachant que le *snake* GVF accentue la marge d'attraction des contours, et comme la carte contour (Fig. IV.8) met en évidence des contours à l'intérieur de la zone d'intérêt, cela explique que le *snake* est attiré par ce contour car il est initialisé proche de lui. Cela explique, aussi, le contour détecté avec l'initialisation automatique (Fig. IV.6(c)).

Cependant, pour les *snakes* initialisés automatiquement, la détection est presque la même pour tous les modèles. Elle est très satisfaisante. Néanmoins, pour les concavités, nous remarquons qu'elles sont très bien détectées par le *snake* GGVF par rapport aux autres *snakes*.

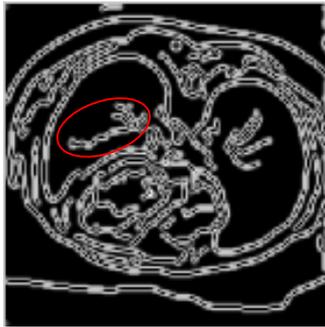


Fig. IV.8 : la carte contour de l'image test.

Dans le tableau IV.1, nous donnons les différents temps d'exécutions de notre programme. Nous pouvons, très bien, constater que les temps d'exécutions, pour l'initialisation automatique, sont réduits de plus de 50% par rapport à l'initialisation manuelle. La cause venait du fait que l'initialisation est très proche du contour initial, ce qui a permis de réduire le nombre d'itérations, et de ce fait d'accélérer la convergence du *snake* vers les contours recherchés. Ajoutant le fait que le processus d'initialisation automatique est basé sur des traitements de bas niveau, qui sont simples est rapides. Le temps total d'exécution de notre programme est de : $t=14,813$ secondes.

<i>snake</i> \ initialisation	Initialisation manuelle	Initialisation automatique
<i>snake</i> classique	77.677	36.702
<i>snake</i> GVF	84.489	40.665
<i>snake</i> GGVF	83.222	42.811

Tableau IV.1 : les différentes estimations des temps (en seconde) d'exécutions du programme sur l'image *chest*.

Nous pouvons, aussi, constater que le temps de convergence du *snake* classique est moins important par rapport aux autres dans les deux cas. Cela s'explique par le temps pris pour le calcul du champ GVF ou GGVF.

IV.3 Application de l'approche sur une image IRM

Notre objectif est de détecter le contour d'une tumeur dans une image IRM. A cet effet, nous avons choisi une image IRM cérébrale, que nous avons récupéré du site [«http // www .STI biotechnologie-ST2S/imagerie médicale et endoscopie/programme de terminal](http://www.STI_biotechnologie-ST2S/imagerie_medicale_et_endoscopie/programme_de_terminal)

ST2S.fr». C'est une image de dimension 635x635, format JPG, enregistrée par le docteur Fabrice BING, Centre Hospitalier Universitaire de Grenoble, France. Elle représente la coupe axiale cérébrale d'un enfant atteint par une tumeur, sur la fosse cérébrale postérieure (Fig. IV.9).

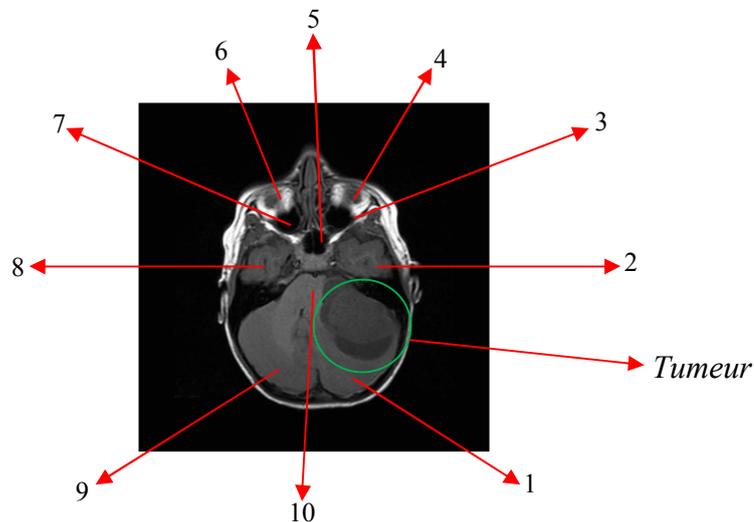


Fig. IV.9 : image IRM du cerveau de la coupe axiale pondération T_1 , avec : 1- lobe occipital droit, 2 - lobe temporal droit, 3 - sinus maxillaire droit, 4 - globe oculaire droit, 5 - sinus sphénoïdal, 6 - globe oculaire gauche, 7 - sinus maxillaire gauche, 8 - lobe temporal gauche, 9 - lobe occipital gauche et 10 - pont.

IV.3.1 La tumeur [16]

La tumeur désigne une prolifération tissulaire ou cellulaire anormale, non inflammatoire, échappant aux mécanismes de régulation de l'organisme. L'expression « *Cancer du cerveau* », également dit cancer ou « *tumeur du système nerveux central* », regroupe plusieurs formes de tumeur susceptibles de se développer dans le cerveau : soit à partir d'une cellule du cerveau lui-même, soit d'une cellule métastatique exportée d'un cancer situé dans une autre partie du corps.

Les tumeurs cérébrales primaires sont, plutôt, situées dans la partie arrière du cerveau, chez les enfants, et dans la partie antérieure des deux tiers des *hémisphères cérébraux*, chez les adultes, mais elles peuvent affecter toutes les parties du cerveau.

IV.3.2 Détection de tumeurs cérébrales [17]

Pour détecter une tumeur sur une image IRM on utilise ce qu'on appelle produit de contraste (Annexe B). Ce produit est introduit par une injection intraveineuse.

Au niveau du cerveau, les parois du système vasculaire cérébral empêchent le produit du contraste de pénétrer à l'intérieur des cellules cérébrales, par contre la tumeur fabrique des vaisseaux dont la paroi est anormale, ce qui permet au produit de contraste de pénétrer à l'intérieur des cellules tumoral

L'accumulation locale de ce produit, dans la tumeur, va apparaître comme un signal hyper intense en pondération T₁ (Annexe B), rendant ainsi visible la tumeur (Fig. IV.10).



Fig. IV.10 : image IRM du cerveau de la coupe axiale pondération T₁, après injection du produit de contraste : tumeur fosse cérébrale postérieure.

Le produit de contraste utilisé, dans ce cas, est le gadolinium (Annexe B). Sa présence permet d'améliorer le contraste des images obtenues et de mettre en évidence des lésions, notamment les tumeurs.

Une IRM pondération T₁ permet de visualiser les tissus (organes et os) par un signal hyper-intense, et les liquides par un signal hypo-intense. Par contre, les IRM pondération T₂ (Annexe B) permettent de visualiser les liquides par un signal hyper-intense et les tissus par un signal hypo-intense. Comme le gadolinium agit, beaucoup plus, sur les liquides, une IRM pondération T₂ donne un contraste plus élevé pour la zone infectée (Fig. IV.11). Pour cela, nous allons appliquer notre approche sur cette image, afin de détecter les contours de la tumeur.

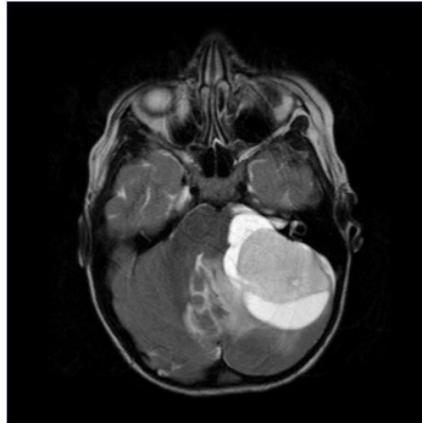


Fig. IV.11 : image IRM du cerveau à coupe axiale pondération T_2 , après injection du produit de contraste : tumeur fosse cérébrale postérieure.

IV.3.3 Initialisation manuelle

Comme pour l'image test, les figures IV.12, IV.13 et IV.14, montrent le résultat de détection des contours de la tumeur dans l'image IRM, par initialisation manuelle d'un *snake* classique, *snake* GVF et *snake* GGVF, respectivement.

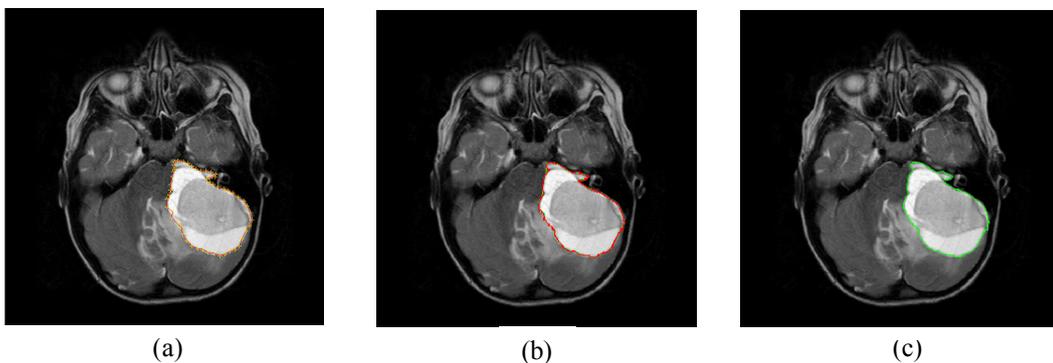


Figure IV.12 : test du *snake* classique : (a) image originale avec initialisation manuelle, (b) déformations et (c) résultat après 80 itérations.

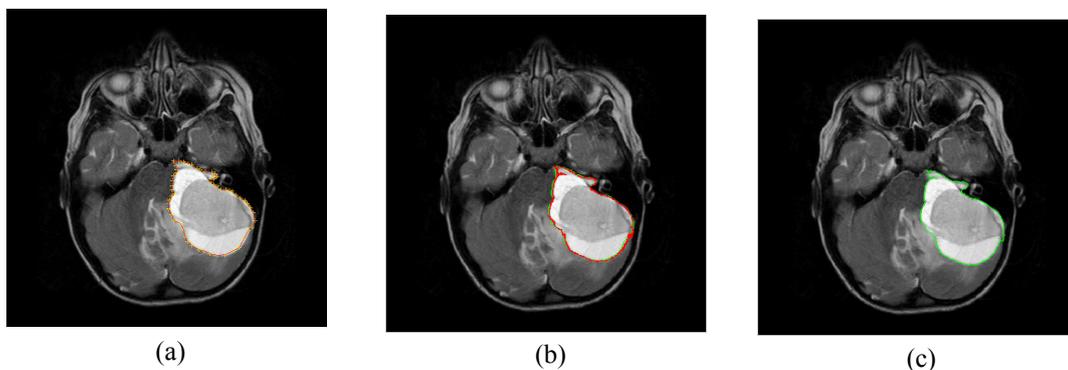


Fig. IV.13 : application du *snake* GVF : (a) image IRM avec initialisation manuelle, (b) déformations et (c) résultat après 80 itérations

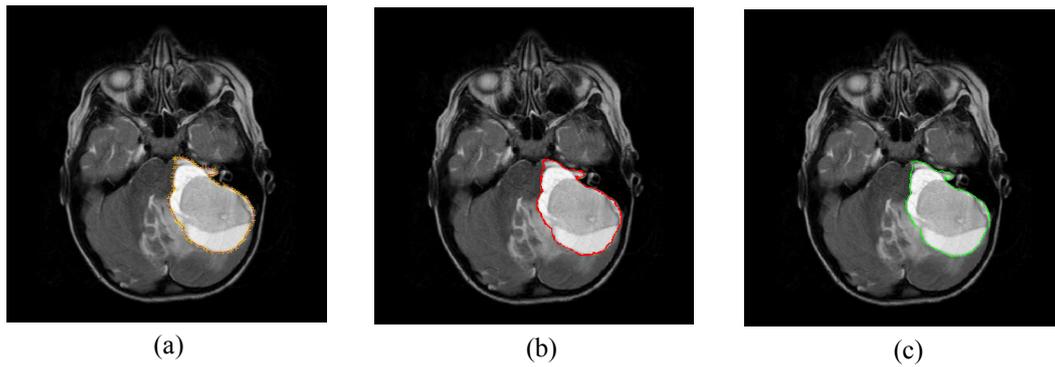


Fig. IV.14 : application du *snake* GGVF : (a) image IRM avec initialisation manuelle, (b) déformations et (c) résultat après 80 itérations.

IV.3.4 Initialisation automatique

Les figures IV.15, IV.16 et IV.17 montrent le résultat de détection des contours de la tumeur dans l'image IRM, par initialisation automatique d'un *snake* classique, *snake* GVF et *snake* GGVF, respectivement.

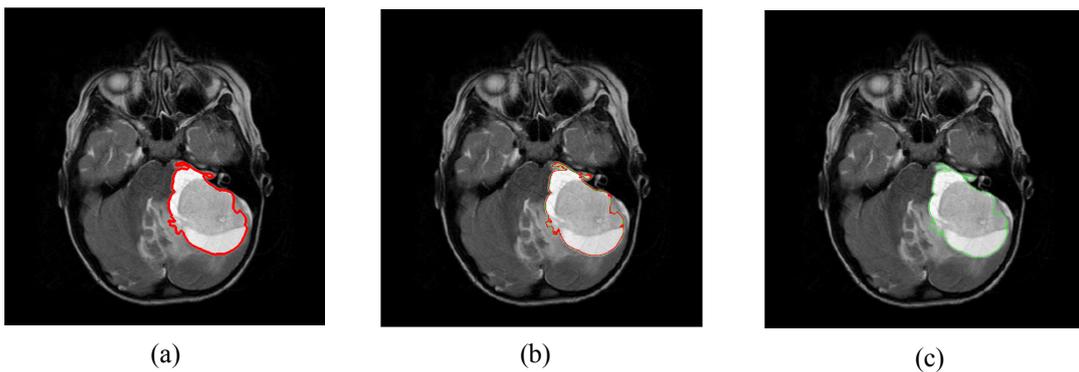


Fig. IV.15 : application du *snake* classique : (a) image originale avec initialisation automatique, (b) déformations et (c) résultat après 10 itérations.

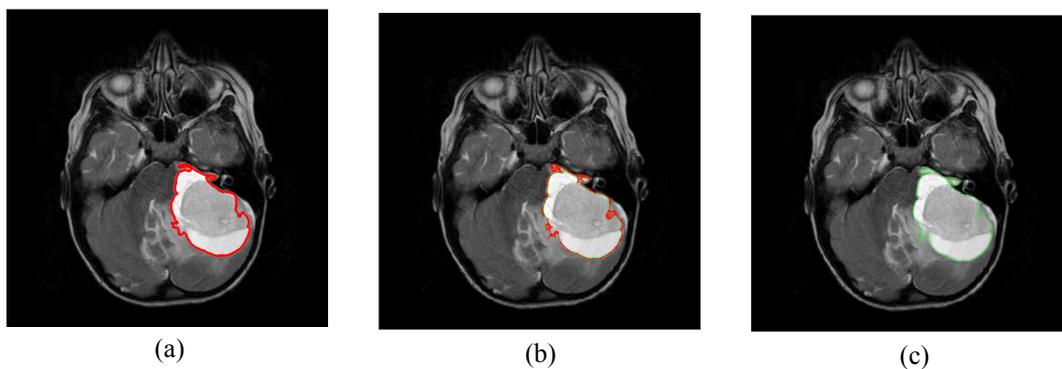


Fig. IV.16 : application du *snake* GVF : (a) image IRM avec initialisation automatique, (b) déformations et (c) résultat après 10 itérations.

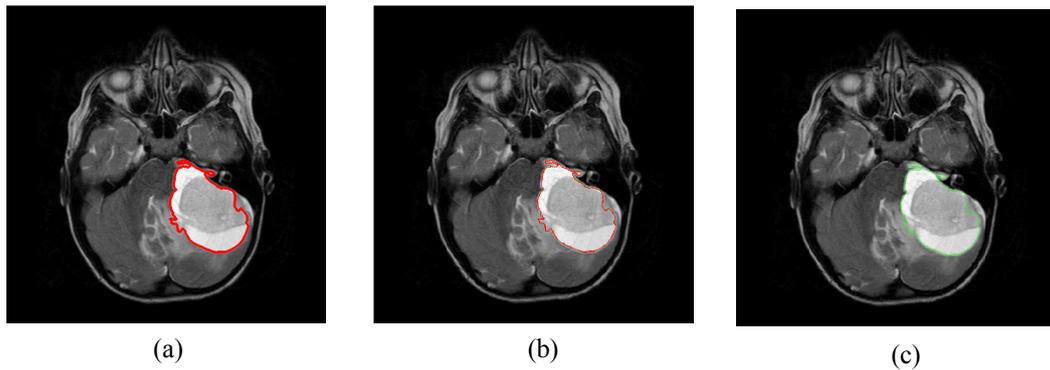


Fig. IV.17 : application du *snake* GGVF : (a) image IRM avec initialisation automatique, (b) déformations et (c) résultat après 10 itérations.

IV.3.5 Interprétation des résultats

On remarque que dans l'initialisation manuelle les cavités de la partie supérieure, de la tumeur, sont mal détectées. Pour le *snake* classique cela est attendu, car si nous observons l'évolution de la courbe dans le temps, nous constatons qu'il commence par se contracter sous l'effet de l'énergie interne qui empêche les irrégularités, pour se fixer, enfin, sur les frontières grâce à l'énergie externe, qui n'est pas assez intéressantes dans ce modèle.

Cependant, pour le *snake* GVF et le *snake* GGVF, la cause revient, probablement, à la multitude des contours dans cette région, ce qui crée des forts gradients piégeant, ainsi, le *snake*. D'ailleurs, nous pouvons le constater sur la carte contour (Fig. IV.18).

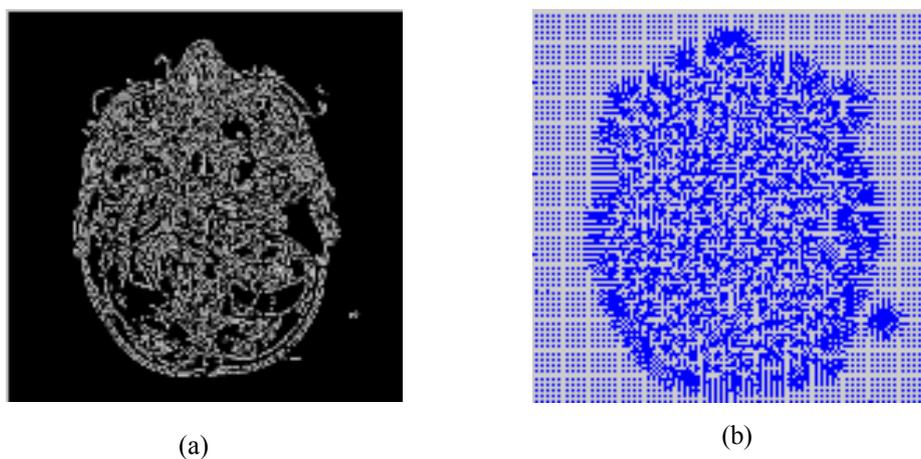


Fig. IV.18 : (a) carte contour et (b) champ GVF.

Pour l'initialisation automatique, les trois *snakes* détectent bien la tumeur. Nous remarquons que le trait en niveau du gris foncé a été pris en considération (la détection est arrêtée à ce niveau) (Fig. IV.19). Cependant, nous pouvons affirmer que les cellules appartenant à ce trait ne sont pas infectées. Donc, la détection est correcte.

Le *snake* GVF a détecté des contours qui sont à l'intérieur de la tumeur (Fig. IV.16(c)), comme c'était le cas pour l'image test (Fig. IV.6 (c)).



Fig. IV.19 : mise en évidence de la ligne en niveau de gris foncé.

Le tableau IV.2, donne les différents temps d'exécution du programme sur l'image IRM. Nous remarquons qu'ils sont, considérablement, élevés par rapport à ceux enregistrés lors de l'exécution du programme sur l'image test. La cause revient, sûrement, au fait que la dimension de l'image IRM est supérieure par rapport à celle de l'image test (635x635 pour l'IRM et 211x225 pour l'image test).

Pour le type d'initialisation, initialisation automatique ou initialisation manuelle, nous constatons la même chose que précédemment (section IV.2.3). Le temps total d'exécution du processus d'initialisation automatique sur l'image IRM est de : **t= 62.143 secondes.**

<i>snake</i> \ initialisation	Initialisation manuelle	Initialisation automatique
<i>snake</i> classique	248.015	119.433
<i>snake</i> GVF	314.357	151.601
<i>snake</i> GGVF	298.715	168.439

Tableau IV.1 : les différentes estimations des temps (en seconde) d'exécutions du programme sur l'image IRM.

IV.4 Conclusion

Tout d'abord, il est intéressant de remarquer que dans le cas d'une initialisation manuelle, les différents résultats se sont avérés médiocres. Nous n'avons jamais réussi à faire une initialisation très proche du contour de la tumeur, ce qui montre bien la nécessité d'une initialisation automatique.

Nous remarquons que la méthode de détection de contour que nous avons développé, permet de s'affranchir d'une initialisation manuelle. La méthode permet d'obtenir une détection correcte du contour en 2 à 3 minutes, sur une image IRM sous MATLAB.

L'originalité de notre approche réside dans l'automatisation de l'initialisation par des prétraitements de faibles complexités, donc rapide. Cette initialisation automatique fournit un contour initial proche et ayant la forme du contour réel, augmentant, ainsi, la convergence. Mais aussi, dans le calcul de la carte contour, que nous avons effectué sur l'image des contours détectés par le filtre de *Canny*.

La conjugaison de ce processus avec les forces GVF fournit un outil de détection de contour rapide et précis.

Conclusion générale



Conclusion générale

Dans notre étude, nous avons au préalable, explicité les différents outils de base du traitement d'image, telle que le filtrage, le seuillage, etc., afin d'améliorer la qualité de l'image. Puis, nous avons discuté quelques approches de détection de contours. Ensuite, nous avons présenté quelques méthodes de segmentation d'images, en s'intéressant aux méthodes basées sur la détection de contours. De ce fait, nous avons discuté la méthode des contours actifs. Nous avons modélisé le *snake*, résolu les équations aux déplacements et construit une procédure de déformation. Pour améliorer la force externe, nous avons présenté la force GVF et la force GVF généralisée.

Nous avons élaboré une méthode de segmentation d'images basée sur les contours actifs classiques, sur les *snake* GVF et GGVF, avec un programme d'initialisation manuelle. Aussi, nous avons apporté une amélioration sur le calcul de la carte contour en se basant sur l'image des contours détectés par le filtre de *Canny*. La difficulté de cette méthode est que le modèle doit être initialisé très proche des contours de l'objet à détecter, ce qui rend la tâche très difficile.

Pour remédier à ce problème, nous avons développé une méthode d'initialisation automatique, fondée sur des prétraitements de faibles complexités. Cette méthode ne requiert qu'une intervention minimale de l'opérateur consistant simplement à choisir sur l'histogramme les deux seuils délimitant la région d'intérêt. Cette initialisation automatique fournit un contour initial proche et ayant la forme du contour réel, augmentant, ainsi, la convergence. Les résultats qui en découlent pour la détection des contours d'une tumeur cérébrale sont satisfaisants, en termes de précision, comparés à ceux obtenus pour l'initialisation manuelle.

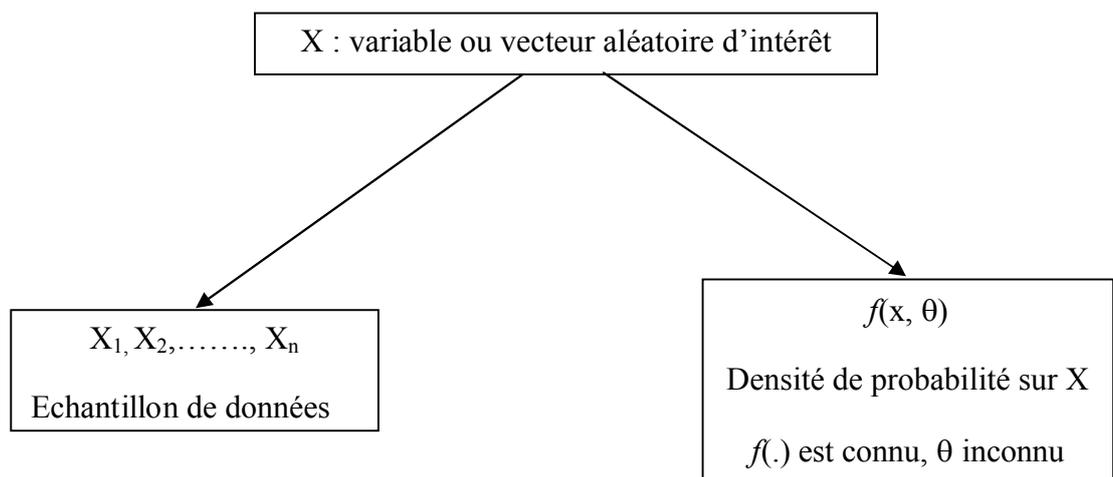
À travers de ce travail, nous nous sommes initiés au domaine du traitement d'images, aussi, au langage de programmation MATLAB. Nous avons appris au travers comment procédés pour mener une recherche.

Enfin, nous espérons avoir apporté un travail qui pourra servir à d'autres finalités.

Annexe A

Estimateur du maximum de la vraisemblance

A.1 But de l'estimation en statistique



L'estimation a pour but de trouver les valeurs possibles de θ telles que la densité $f(x, \theta)$ s'ajuste le mieux aux données disponibles.

Pour effectuer l'estimation, on trouve plusieurs estimateurs : estimateur des moments, estimateur des moindres carrés, estimateur du maximum de vraisemblance.

A.2 Estimateur du maximum de vraisemblance

A.2.1 Fonction de vraisemblance

Soit X une variable aléatoire de densité de probabilité $f(x, \theta)$, où θ est un vecteur de K paramètres : $\theta = (\theta_1, \theta_2, \dots, \theta_k)$.

Soit : X_1, X_2, \dots, X_n un échantillon de données indépendantes.

La fonction de vraisemblance associée est définie par :

$$L(\theta) = \prod_{i=1}^n f(x_i, \theta) \quad (\text{A.1})$$

L'estimateur de maximum de vraisemblance de θ est la valeur de θ qui maximise la fonction de vraisemblance $L(\theta)$.

En pratique on maximise le logarithme de la fonction de vraisemblance :

$$l(\theta) = \ln(L(\theta)) = \sum_{i=1}^n \ln(f(x_i, \theta)) \quad (\text{A.2})$$

A.2.2 Ajustement d'une distribution normale (approche analytique)

La densité d'une distribution normale est définie par :

$$f(x, \theta) = f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \quad (\text{A.3})$$

Avec : $\theta(\mu, \sigma^2)$.

La fonction de vraisemblance associée est :

$$L(\theta) = \prod_{i=1}^n f(x_i, \theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\mu)^2/2\sigma^2} \quad (\text{A.4})$$

Le logarithme de vraisemblance est donc :

$$l(\theta) = \sum_{i=1}^n \ln(f(x_i, \theta)) = -\frac{n}{2} \ln(\sigma^2) - \frac{n}{2} \ln(2\pi) - \sum_{i=1}^n \frac{(x_i-\mu)^2}{2\sigma^2} \quad (\text{A.5})$$

La maximisation revient à chercher les valeurs des paramètres qui annulent la dérivée de $l(\theta)$ donc :

$$\frac{\partial}{\partial \mu} l(\theta) = \sum_{i=1}^n 2 \frac{(x_i-\mu)}{2\sigma^2} = 0 \Rightarrow \hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{X} \quad (\text{A.6})$$

$$\frac{\partial}{\partial \sigma^2} l(\theta) = -\frac{n}{2\sigma^2} + \sum_{i=1}^n \frac{(x_i-\mu)^2}{2\sigma^4} = 0 \Rightarrow \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2 \quad (\text{A.7})$$

L'estimateur de maximum de vraisemblance obtenue est :

$$\hat{\mu} = \bar{X} \quad ; \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})^2 \quad (\text{A.8})$$

Si une approche numérique de dérivation est difficile à résoudre, on utilise les méthodes numériques d'optimisation de fonctions, tel est le cas pour un mélange de gaussiennes.

A.2.3 Ajustement d'un mélange de gaussiennes (approche numérique)

Un mélange de gaussiennes est défini par la somme pondérées de plusieurs gaussiennes :

$$G(x, \theta_j) = \sum_{j=1}^{\alpha} w_j \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{\left(\frac{-1}{2\sigma_j^2}(x^2 - m_j^2)\right)} \quad (\text{A.9})$$

Ou w_j représente le poids de chaque gaussiennes et $\theta_j(m_j, \sigma_j^2)$ le vecteur des paramètres de la gaussiennes j .

Alors la fonction de vraisemblance associée est :

$$L(\theta) = \prod_{i=1}^n G(x_i, \theta_j) = \prod_{i=1}^n \sum_{j=1}^{\alpha} w_j \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{\left(\frac{-1}{2\sigma_j^2}(x_i^2 - m_j^2)\right)} \quad (\text{A.10})$$

Le logarithme de la vraisemblance est donc :

$$l(\theta) = \sum_{i=0}^n \ln(G(x_i, \theta_j)) = \sum_{i=1}^n \ln\left(\sum_{j=1}^{\alpha} w_j \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{\left(\frac{-1}{2\sigma_j^2}(x_i^2 - m_j^2)\right)}\right) \quad (\text{A.11})$$

D'où :

$$l(\theta) = \sum_{i=1}^n \prod_j^{\alpha} \left(\ln w_j - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln(\sigma_j^2) - \frac{1}{2\sigma_j^2} (x_i^2 - m_j^2) \right) \quad (\text{A.12})$$

La maximisation, donc, revient à résoudre l'équation suivante :

$$\frac{\partial}{\partial \theta} l(\theta) = 0 \quad (\text{A.13})$$

Une telle équation est difficile à résoudre analytiquement, alors on fait recoure aux méthodes numériques d'optimisation des fonctions

Dans le cas de mélange de gaussiennes, les paramètres sont estimés en utilisant l'algorithme EM (espérance-maximisation).

Cet algorithme estime les paramètres itérativement et chaque itération se divise en deux étapes :

➤ **Etape E** (Espérance)

Cette étape consiste à calculer l'espérance de la vraisemblance complétée conditionnellement aux données observées au vecteur paramètres courant. Alors, cette étape nécessite simplement le calcul de la probabilité *a posteriori* :

$$p^k(j, x_i) = \frac{\omega_j^k p(x_i, \theta_j^k)}{\sum_{j=1}^3 \omega_j^k p(x_i, \theta_j^k)} \quad (\text{A.14})$$

$$\text{Avec : } \left\{ \begin{array}{l} \mathcal{N}(m_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-1}{2\sigma_i^2}(x^2 - m_i^2)\right)} \\ P(x, \theta_i) = \mathcal{N}(m_i, \sigma_i^2) \end{array} \right. \quad (\text{A.15})$$

➤ **Etape M** (Maximisation)

Dans cette étape on calcule le vecteur des paramètres qui maximise l'espérance. Cette maximisation fournit les paramètres :

$$\omega_i^{k+1} = \frac{1}{n} p^k(i, x_j) \quad (\text{A.16})$$

$$m_i^{k+1} = \frac{\sum_{j=1}^n x_j p^k(i, x_j)}{\sum_{j=1}^n p^k(i, x_j)} \quad (\text{A.17})$$

$$(\sigma_i^{k+1})^2 = \frac{\sum_{j=1}^n (x_j - m_i^{k+1})^2 p^k(i, x_j)}{\sum_{j=1}^n p^k(i, x_j)} \quad (\text{A.18})$$

On définit le vecteur des paramètres $\varphi^{k+1} = (\omega_i^{k+1}, m_i^{k+1}, (\sigma_i^{k+1})^2)$.

L'algorithme s'arrête lorsque :

$$\|\varphi^{k+1} - \varphi^k\| \leq \varepsilon \quad (\text{A.19})$$

Annexe B

Principes élémentaires de l'IRM

B.1 Principes physiques élémentaires de l'IRM [19]

Cette section résume les principes élémentaires de l'imagerie par résonance magnétique.

B.1.1 Champ et moment magnétique

Le moment magnétique est une grandeur vectorielle caractéristique d'un circuit électrique qui permet d'étudier son comportement dans un champ magnétique. Ainsi, un électron qui décrit une orbite autour d'un noyau constitue une boucle de courant, de moment magnétique $\vec{\mu}$ perpendiculaire à l'orbite.

Lorsqu'un proton est placé dans un champ magnétique \vec{B}_0 statique, son moment magnétique est animé d'un mouvement de précession (Fig. B.1) autour de l'axe de vecteur directeur \vec{B}_0 et sa fréquence de rotation est exprimée par l'équation (B.1) où γ est le rapport gyromagnétique spécifique au noyau étudié :

$$f_0 = \frac{\gamma}{2\pi} B_0 \quad (\text{B.1})$$

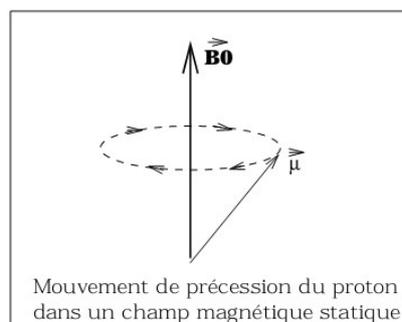


Fig. B.1 : mouvement de précession du proton dans un champ magnétique statique.

Dans le cas du proton, la mécanique quantique n'autorise que deux états possibles de spin $\left(\frac{-1}{2}, \frac{+1}{2}\right)$ $\vec{\mu}$ sera donc orienté soit parallèlement à \vec{B}_0 , soit dans la direction opposée à \vec{B}_0 . Comme la population ayant un spin de $1/2$ est très légèrement supérieure à la population ayant un spin de $-1/2$, le moment magnétique résultant \vec{M} sera orienté dans le sens de \vec{B}_0 . Sa composante en z, \vec{M}_z , est petite devant \vec{B}_0 . Pour mesurer ce moment magnétique, il faut donc le basculer dans le plan transverse XY. Pour cela, on applique une onde radiofréquence (RF) égale à f_0 et perpendiculaire à \vec{B}_0 .

B.2 L'appareil d'IRM

C'est un gros électro-aimant, il est disposé autour d'un cylindre dont la taille est suffisante pour qu'un patient puisse s'y glisser. L'électro-aimant est entouré de 20 à 30 kilomètres de fils hélicoïdaux faits de matériaux super conducteurs qui n'offrent aucune résistance à l'électricité. Ainsi, lorsqu'un courant électrique est introduit dans les fils, il se maintient à pleine vitesse pendant des années sans qu'il soit nécessaire d'ajouter plus d'influx électrique. Plus le nombre d'enroulements est grand, plus le champ magnétique est puissant. La force du champ magnétique se mesure en Tesla. Les appareils d'imagerie clinique disposent d'aimants dont le champ magnétique varie entre 0,02 et 1,5 Tesla. La puissance de ces aimants est telle que tout objet métallique pénétrant le champ électromagnétique (comme une clé) est immédiatement attirée. Pour préserver sa super-conductivité, l'aimant est maintenu dans un froid intense. Les fils hélicoïdaux qui l'entourent sont disposés dans un appareil à double paroi qui baigne dans de l'hélium liquide maintenu à $4,2^\circ\text{C}$, au-dessus du zéro absolu, température à laquelle les molécules s'immobilisent. L'appareil est maintenu dans le vide et placé dans un réservoir rempli d'azote liquide. L'ensemble est placé dans une armure en acier ou en cuivre qui porte le nom de cage de Faraday, qui bloque les signaux de radiofréquence des stations de radio locales et des postes de télévision qui peuvent interférer avec les signaux RM.

B.3 Les atomes dans le champ magnétique

L'atome dispose d'un noyau et d'électrons qui tournent autour du noyau. Le noyau est composé de protons et de neutrons. Les protons ont leur propre champ magnétique avec un pôle nord et un pôle sud. Ils tournent autour de leur axe comme la terre. Lorsqu'un patient est introduit dans l'appareil, les protons de son organisme s'alignent dans le même sens que le champ magnétique de l'appareil. Sous contrôle technique, la bobine haute fréquence du

scanner émet des ondes radio qui amènent les protons à se réaligner. Lorsque les ondes radio et les protons vibrent à la même fréquence, les protons absorbent une partie de l'énergie des ondes radio. C'est ce qu'on appelle la résonance et c'est de ce phénomène que dérive la résonance magnétique. Lorsque l'émission de l'antenne s'interrompt, les atomes d'hydrogène reprennent leur position d'équilibre dans le champ magnétique (relaxation) en redonnant de l'énergie. Cette quantité d'énergie peut alors être mesurée grâce à la même antenne. L'énergie redistribuée peut être décomposée suivant deux axes :

Un axe parallèle au champ magnétique ou relaxation longitudinale ou T1.

Un axe perpendiculaire au champ magnétique ou relaxation transversale ou T2.

B.4 Constitution des images

Les relaxations « T1 » et « T2 » des protons dépendent de la nature des tissus. C'est à partir de ce « différentiel » que l'on peut obtenir deux images de contraste différent des différents tissus. Ces images sont appelées images pondérées « T1 » et « T2 ».

En IRM, il n'existe pas d'échelle de densité. On parle alors « d'hyper-intensité » ou « d'hypo-intensité » en « T1 » ou en « T2 ». Cet aspect est défini par rapport à l'aspect d'un tissu quelconque adjacent.

B.5 Produit de contraste

Les éléments à propriétés paramagnétiques, c'est-à-dire possédant dans leurs couches électroniques des électrons célibataires comme le gadolinium, le manganèse et le fer sont utilisés en médecine dans un but diagnostique pour renforcer le contraste en imagerie par résonance magnétique (IRM).

B.5.1 Le gadolinium

Le gadolinium est un métal de haut poids atomique : 157. Il porte le numéro atomique 64. Le gadolinium est un élément classé parmi les terres rares, présent seulement à l'état de traces dans la nature. Il existe sous forme de plusieurs isotopes stables, le plus abondant ayant la masse 158. Du fait de l'existence de sept électrons célibataires, le gadolinium a un effet paramagnétique ou ferromagnétique très élevé.

B.5.2 l'utilité du gadolinium

Les produits de contraste sont utilisés pour rehausser les contrastes. Une particularité intéressante de ces produits hydrosolubles peut cependant être exploitée pour l'exploration de l'encéphale. En effet, il existe normalement une « *barrière hémato-encéphalique* » imperméable aux produits de contraste. Lorsqu'il y a une lésion cérébrale inflammatoire, tumorale ou traumatique, cette « *barrière* » se trouve rompue et le produit de contraste pénètre et stagne dans le tissu interstitiel cérébral.

En raison de son effet paramagnétique, le gadolinium présent dans l'eau accélère les vitesses T1 et T2 de relaxation des protons après leur stimulation par un champ magnétique. L'IRM utilise la propriété des protons de l'organisme, en particulier ceux de l'eau, d'émettre des signaux lorsqu'ils retournent à leur état d'équilibre après avoir été stimulés par un champ magnétique. La présence d'un élément à effet paramagnétique élevé, comme le gadolinium au contact des protons de l'eau, accélère leurs temps de relaxation T1 et T2 et permet d'améliorer le contraste des images obtenues et de mettre en évidence des lésions, notamment les tumeurs.

B.6 Observation du cerveau

L'observation par des coupes en deux dimensions du cerveau peut être effectuée selon plusieurs angles de vue. Ainsi, on distingue trois axes anatomiques pour réaliser les coupes (Fig. A.3).

1. **Coupes axiales** : Ces coupes correspondent quasiment à un plan horizontal. En imagerie de résonance magnétique, elles correspondent à un plan perpendiculaire à l'axe du champ magnétique principal.
2. **Coupes sagittales** : Ces coupes sont prises dans des plans parallèles au plan inter-hémisphérique. Il s'agit de vues latérales du cerveau.
3. **Coupes coronales** : Ce sont des coupes perpendiculaires aux coupes axiales et sagittales. Elles correspondent à des vues de face du cerveau.

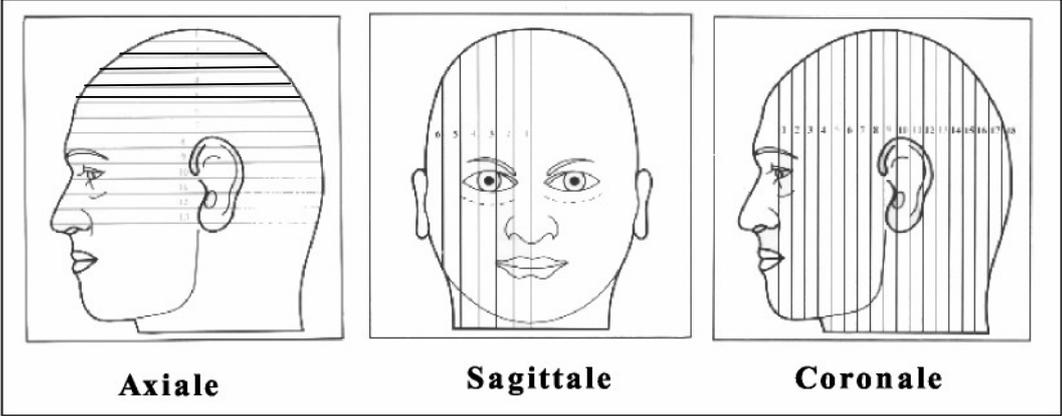


Fig. B.2: les trois axes de coupe pour la visualisation du cerveau.

Bibliographie

- [1] : **Lounaci D et Lazib R**, « traitement des images Meteosat de format 'OpenMTP' », Mémoire d'ingénieur, Département Electronique, UMMTO, 2004.
- [2] : **Houmel A et Harbane D**, « segmentations d'image en niveau de gris par fusion de région », Mémoire d'ingénieur, Département Electronique, UMMTO, 2005.
- [3] : **Jérémy L et Christian B**, « Segmentation d'images cérébrales : État de l'art », Rapport de recherche N° 6306, Rennes, France, Juillet 2007. Téléchargé du site « [http // www.inria.fr](http://www.inria.fr) », dernière visite 05/08/2008.
- [4] : **Bloch I et Gousseau Y**, « Le traitement des images », tome 2, Polycopié du cours ANIM, Département TSI - Télécom-Paris, septembre 2004. Téléchargé du site « [http // www.cmse.fr/fr/transfert /spin / formation / ressources/ sam96/ fichier pdf / animage .pdf](http://www.cmse.fr/fr/transfert/spin/formation/ressources/sam96/fichier/pdf/animage.pdf) », dernière visite 05/08/2008.
- [5] : Document internet « Analyse d'image », téléchargé du site
- [6]: **Pecaud C, Lyncee J-L et Dess G**, « Seuillage et segmentation d'images texturées »,
- [7] : **Bidi Georges E-F**, « Segmentation par seuillage d'images satellitaires METEOSAT», Mémoire d'ingénieur, Département Electronique, UMMTO, 2004.
- [8] : **Rousselle J J**, « Les contours actifs, une méthode de segmentation. Application a l'imagerie médicale», Thèse Doctorat, université de François Rabelais, Tours, France, juillet 2003. Téléchargé du site, « [http // www-Isis.enst.fr/rilk/gdr/kiosque/thèse HDR](http://www-Isis.enst.fr/rilk/gdr/kiosque/thèse HDR) », dernière visite.02/09/2008.
- [9] : **Burel G**, « Introduction au traitement d'images », Simulation sous Matlab
- [10] : **Burel M et Obert C**, «traitement d'image TP N° 6 : Morphologie Mathématique »,

Institut National des sciences Appliquées. ROUEN. France, 16/05/2005.

[11] : **Schwartz P**, « Méthode des contours actifs », Publication, mars 2008. Téléchargé du site « <http://khayyam.developpez.com/articles/algo/contours-actifs/> ».derniere visite 2/09/2008.

[12] : **Chilali. O**, « Classification automatique de données utilisant les modèles déformables », Mémoire de magister, Département d'Automatique, UMMTO, 2006.

[13] : **Piron. V**, « Segmentation du myocarde par modèle déformable en IRM du petit animal », Thèse DEA, université d'Angers, France, 2002.

[14]: **XU C**, « Deformable Models With Application To Human Cerebral Cortex Reconstruction From Magnetic Resonance Images», Thèse de doctorat, Université Johns Hopkins, Baltimore, Maryland, USA, janvier 1999.

[15]: **XU C et PRINCE J-L**, « Snakes, Shapes And Gradient Vector Flow »,IEEE Transaction on Image Processing, 1998.

[16]: **document internet.**

« www.prostatecancer.ca/french/prostate_owners_manual/living/glossary/ », dernière visite. 20/10/2008.

[17]: **document internet.**

« [www.atlas-IRM/IRM axial du cerveau.com](http://www.atlas-IRM/IRM_axial_du_cerveau.com) », dernière visite 20/10/2008.

[18]: **Martin. B**, « Etude statistique d'histogrammes en image ».rapport de stage, Master MAD 1^{ère} année, université d'Orléans, France, 2005/2006.téléchargé du site, « <http://alainparaponaris.free.fr/stageisto/> », dernière visite 02/09/2008.

[19] : **document internet**

« [www. Info cancer .fr](http://www.Info_cancer.fr) », dernière visite 10/10/2008.