

*République Algérienne Démocratique et Populaire*  
*Ministère de l'Enseignement supérieur et de la recherche scientifique*  
*Ecole Nationale Polytechnique.*



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

**DEPARTEMENT D'ELECTRONIQUE**

**MEMOIRE DE MAGISTER**

Présenté par :

**Mme Hamidatou née Hamadi Fatma Zohra**  
**Ingénieur d'Etat en Electronique (USD de Blida)**

**Intitulé**

**Techniques de segmentation d'images par contours actifs  
et contribution au mouvement dans des séquences  
d'images.**

**Devant le jury d'examen composé de :**

**Présidente : M. GUERTI                      Maître de conférences                      ENP**

**Rapporteur : L. HAMAMI                      Maître de conférences                      ENP**

**Examineurs : B. BOUSSEKSSOU                      Chargé de cours                      ENP**

**S. AIT DAOUD                      Chargée de cours                      INI**

**C. FIALA                      Chargée de cours                      U. Boumerdés**

**30 Octobre 2007**

# Remerciements

*Je tiens à remercier très sincèrement ma promotrice Madame Latifa Hamami, qui a su croire en moi, et dont le soutien, le suivi, la disponibilité, et les conseils m'ont été plus que précieux.*

*Mes remerciements vont également aux enseignants membres de jury de ce travail.*

*Enfin, mes remerciements à toute personne ayant contribué de près ou de loin à l'aboutissement de ce modeste travail.*

# *Dédicaces*

*Je dédie ce travail à la mémoire de mes parents ;*

*A mon mari*

*A mes enfants*

*A toute ma famille.*



# SOMMAIRE

Introduction Générale.....	1
<b>Chapitre 1: Méthodes de traitement d'images</b>	
1 Introduction .....	3
2 Prétraitement .....	3
2.1 Réduction de bruit.....	4
2.1.1 Filtres linéaires.....	4
2.1.2 Filtres non linéaires.....	5
2.2 Modification d'histogramme.....	7
2.3 Rehaussement du contraste.....	7
3. Segmentation.....	8
3.1 Introduction .....	8
3.2 Définition formelle de la segmentation.....	8
3.3 Les approches de la segmentation.....	9
3.3.1 Approche contours(frontières).....	9
3.3.2 Approche région .....	13
3.3.3 Autres approches.....	15
<b>4. Conclusion.....</b>	<b>15</b>
<b>Chapitre 2: Contours actifs et algorithmes génétiques</b>	
1. Introduction.....	16
2. Historique des modèles déformables.....	16
2.1 Qu'est ce qu'un contour actif ?.....	17
2.2 Principe des contours actifs.....	18
3. Les énergies.....	20
3.1 Energie interne.....	20
3.2 Energie externe.....	20
3.3 Energie de contexte.....	21
4. Méthodes d'optimisation.....	21
4.1 Approche variationnelle.....	22
4.2 Algorithme de Greedy.....	26
5 Algorithmes génétiques.....	28
5.1 Introduction.....	28

5.2 Le codage.....	29
5.3 La fonction d'adaptation.....	30
5.4 Les opérateurs de reproduction .....	31
conclusion .....	35

### Chapitre 3 : Suivi d'objets en mouvement

1. Introduction.....	36
Mouvement (déplacement).....	36
2.1 Mouvement d'un objet rigide.....	36
2.2 Mouvement d'un objet déformable (non rigide).....	36
3. Estimation de mouvement.....	37
4. Segmentation par le mouvement.....	38
5. Suivi par contour actif (snake).....	38
conclusion .....	40

### Chapitre 4 : Conception

1. Introduction.....	41
2. Transformation de l'image en niveaux de gris.....	41
3. Prétraitement, lissage de l'image.....	41
4. Contour actifs et algorithme de Greedy.....	43
4.1 Initialisation du contour.....	43
4.2 Prétraitement.....	45
4.3 Minimisation et déformation par algorithme de Greedy.....	45
4.3.1 Calcul des énergies.....	45
4.3.2 Parcours des différents paramètres supplémentaires réglables.....	46
4.3.3 Mouvement du contour.....	47
4.3.4 Extraction de la courbe.....	48
5. Contours actifs et algorithme du Greedy autonome.....	49
5.1 Minimisation et déformation par algorithme du Greedy autonome.....	49
6. Suivi par contour actif.....	50
6.1 Initialisation de contour sur la première image.....	52
6.2 Réinitialisation sous forme de rectangle.....	52
6.3 Déformation du contour actif.....	53
7. Implémentation par algorithme génétique.....	54
7.1 Les différentes étapes de l'algorithme.....	55
7.1.1 Initialisation.....	55

7.1.2 Evaluation.....	56
7.1.3 Parent.....	56
7.1.4 Les opérateurs génétiques.....	57
7.1.5 Meilleur individu.....	57
Conclusion.....	58

## Chapitre 5 : Mise en œuvre

1. Introduction.....	59
2. Interface du logiciel.....	59
2.1 Description de la première partie de l’interface.....	59
2.1.1 Les menus .....	59
2.1.2 Description de la première partie de l’interface (image statique).....	60
2.1.2.1 Module Greedy.....	60
2.1.2.2 Module Greedy autonome.....	62
2.1.2.3 Module algorithmes génétiques.....	62
2.1.2.4 Module de comparaison.....	63
2.1.3 Description de la deuxième partie de l’interface (image dynamique).....	64
2.1.3.1 Pour le suivi par Greedy.....	64
2.1.3.2 Pour le suivi par Greedy autonome.....	65
3. Conclusion.....	66

## Chapitre 6 : Tests et résultats

1. Introduction.....	67
2. Cas statique.....	67
a) Résultat pour le cercle.....	68
b) Résultat pour le nuage.....	69
c) Résultat pour le carré.....	70
d) Résultat pour l’étoile .....	72
e) Résultat pour le triangle .....	73
3. Cas dynamique (séquences d’images).....	74
3.1 Résultat pour Greedy .....	74
3.2 Résultat pour Greedy autonome.....	76
4. Conclusion.....	77
Conclusion générale.....	78

## Bibliographie

# Liste des figures

Figure 2.1 : Types de contours actifs.....	18
Figure 2.2 : Principe des contours actifs.....	19
Figure 2.3 : Principe de base d'un algorithme génétique.....	29
Figure 2.4 : La roulette.....	32
Figure 2.5 : le tournoi en deux individus avec rééligibilité.....	32
Figure 2.6 : Le croisement d'un point.....	33
Figure 2.7 : Croisement uniforme.....	34
Figure 2.8 : La mutation avec le codage binaire.....	34
Figure 2.9 : La mutation avec codage par valeurs entières.....	35
Figure 3.1 : Exemple : boule de pétanque et boule anti-stress.....	37
Figure 4.1 : Masque du filtre Moyen.....	41
Figure 4.2 : Exemple de lissage par un filtre moyen 3x3.....	41
Figure 4.3 : Masque de Sobel.....	41
Figure 4.4 : Organigramme de l'algorithme de Greedy.....	43
Figure 4.5 : Les deux points représentant la courbe initiale.....	42
Figure 4.6 : Exemple d'un voisinage de 3.....	45
Figure 4.7 : Algorithme de la déformation par la méthode de Greedy.....	47
Figure 4.8 : Algorithme pour le contour actif autonome.....	48.
Figure 4.9 : Organigramme de suivi d'objet en mouvement par des contours actifs.....	50
Figure 4.10 : Algorithme de réinitialisation automatique pour Greedy (resp Greedy autonome).....	51
Figure 4.11 : Procédure générale pour le traitement des séquences d'images.....	52
Figure 5.1 : Fonction du module Greedy.....	59
Figure 5.2 : Module de Greedy simple (image statique).....	60
Figure 5.3 : Fonction du module Greedy autonome.....	61
Figure 5.4 : Module de Greedy autonome (image statique).....	61
Figure 5.5 : Fonction du module algorithme génétique.....	62
Figure 5.6 : Module algorithme génétique.....	62
Figure 5.7 : Module de comparaison.....	63

Figure 5.8 : Module de suivi par l'algorithme de Greedy.....	64
Figure 5.9 : Module de suivi par l'algorithme de Greedy autonome.....	65
Figure 6.1 : Tableau de comparaison entre Greedy, Greedy autonome et algorithme génétique.....	66
Figure 6.2 : Résultat obtenu par Greedy simple (cercle).....	67
Figure 6.3 : Résultat obtenu par Greedy autonome (cercle).....	67
Figure 6.4 : Résultat obtenu par l'algorithme génétique (cercle).....	68
Figure 6.5 : Résultat obtenu par Greedy simple (étoile).....	68
Figure 6.6 : Résultat obtenu par Greedy autonome (étoile).....	69
Figure 6.7 : Résultat obtenu par l'algorithme génétique (étoile).....	69
Figure 6.8 : Résultat obtenu par Greedy simple (carré).....	69
Figure 6.9 : Résultat obtenu par Greedy autonome (carré).....	70
Figure 6.10 : Résultat obtenu par l'algorithme génétique (carré).....	70

## INTRODUCTION GENERALE

Les images font partie intégrante de la vie humaine, nous synthétisons des images à des fins artistiques. Nous générons et interprétons des images à des fins scientifiques (images médicales par exemple). Dans toutes ces activités la vision par ordinateur est un outil puissant. C'est pourquoi il paraît crucial de doter les systèmes robotiques de l'outil de la vision.

Le traitement d'images par contours actifs est le domaine qui nous préoccupe ici particulièrement ; il présente un intérêt dans quasiment tous les domaines où le problème de la recherche automatique d'information dans des images s'avère un besoin.

Pour la surveillance, l'analyse de séquences d'images pose de nombreux problèmes, tels le suivi d'objets, de personnes [LEF, 01], le suivi de cibles pour les militaires ou en robotique [ALE, 97]. Tout cela débouche souvent sur des problèmes de reconnaissance de formes : reconnaissance de visages [RAD, 95], d'empreintes digitales, d'iris de l'œil, de textes, de dessins etc.

La première étape dans le processus de compréhension d'une image est le plus souvent l'extraction des objets qui la composent. Aussi, la segmentation est primordiale en analyse d'images, du simple seuillage des niveaux de gris aux techniques plus complexes. Nous nous sommes intéressées à la méthode des contours actifs. C'est une technique particulière de segmentation. Son principal avantage est de fournir un contour consistant en une chaîne a priori de points. Cet avantage est primordial car il évite le recours à des algorithmes de suivi de contour dans lesquels la question de l'appartenance ou non de certains pixels au contour se pose toujours. En effet, les points de contour constituent rarement des courbes fermées et une étape de fermeture est nécessaire. Mais toute avancée crée de nouveaux problèmes. Tout d'abord, il existe maintenant de nombreux paradigmes dans la famille des modèles déformables. Il y aura donc à effectuer un choix du modèle. Il sera basé sur des critères de rapidité d'exécution, de complexité d'implémentation, de type d'élément à segmenter, de type d'image à traiter. Nous avons arrêté notre choix sur l'implémentation des algorithmes de Greedy et de Greedy autonome pour leur critère de rapidité d'exécution puis l'implémentation par les algorithmes génétiques pour leur rigueur. Nous ferons enfin une comparaison des trois algorithmes cités selon leurs temps d'exécution et le nombre d'itérations effectuées.

Comme le suggère le thème de notre mémoire « Techniques de segmentation par contours actifs et contribution au mouvement dans des séquences d'images », Ce dernier a été organisé en deux parties :

Les trois premiers chapitres relatent la première partie dans laquelle le chapitre 1 est réservé aux méthodes de traitement d'images où sont présentées quelques techniques de segmentations d'images ; le chapitre 2 est consacré aux contours actifs détaillant l'approche variationnelle ainsi que d'autres techniques parmi les plus utilisées . On y trouve aussi une introduction aux algorithmes génétiques ; le chapitre 3 quant à lui présente le suivi d'un objet en mouvement dans une séquence d'images.

Les deux chapitres suivants constituent la deuxième partie. Le chapitre 4 présente la démarche suivie en expliquant les différents acheminements menant à la mise en œuvre des algorithmes implémentés.

A travers le chapitre 5 nous présentons les résultats obtenus après une série de tests ainsi que l'interprétation de ces résultats.

Une conclusion générale fera le point de ce travail et donnera quelques perspectives envisagées.

# Chapitre 1

## Méthodes de Traitement d'Images

### 1. Introduction

Il est parfois demandé à l'homme d'être minutieux dans son travail dans lequel l'exactitude des résultats obtenus est primordiale, et même parfois vitale. Dès lors ; l'introduction du traitement d'images dans différents secteurs où l'homme opère, est nécessaire pour remplacer l'observateur humain par la machine, ou au moins l'aider dans quelques tâches, étant donné que l'ordinateur permet une meilleure qualité et une excellente visibilité.

Les méthodes de traitement des images sont nombreuses, toutefois elles peuvent être réparties en deux modules majeurs, à savoir celui englobant les méthodes de prétraitement et celui destiné aux méthodes de segmentation [BLO, 05]. Nous pouvons les définir de la façon suivante :

1. Les méthodes de prétraitement telles que :

- Le Changement de contraste et de luminosité (amélioration de la qualité);
- Le filtrage convolutif pour éliminer le bruit.

2. Les méthodes de segmentation telles que :

- Les méthodes de détection de contours. Comme les méthodes dérivatives ou le filtrage optimal;
- Les méthodes de segmentation en régions homogènes. Comme le seuillage ou la croissance de régions.

### 2. Prétraitement

En amont de la segmentation on retrouve l'étape du prétraitement. C'est une étape qui a pour but de faciliter la segmentation en accentuant les ressemblances des pixels d'une même région, ou les dissemblances entre les pixels appartenant à des régions distinctes.

Dans ce qui suit nous allons présenter quelques méthodes de prétraitement.

## 2.1. Réduction de bruit

La diminution du bruit a pour but de réduire l'amplitude des variations d'intensités à l'intérieur d'une même région sans porter atteinte aux zones de transitions qui définissent les frontières entre les régions.

La réduction du bruit est souvent réalisée par une opération de filtrage, qui permet en même temps d'améliorer la qualité de l'image.

Le principe du filtrage est l'application d'une opération au niveau du pixel élémentaire en tenant compte de son environnement (pixels voisins).

C'est une opération qui consiste à déplacer un filtre (masque), qui est une matrice de dimension impaire par exemple  $3 \times 3$ ,  $5 \times 5$ , sur l'image suivant chaque pixel et de remplacer la valeur de ce dernier par le résultat de l'opération effectuée sur ses voisins. A cet effet ; plusieurs filtres ont été définis :

- Les filtres linéaires ;
- Les filtres non linéaires.

### 2.1.1. Filtres linéaires

Le filtre linéaire fait en sorte que chaque pixel de l'image soit remplacé par une valeur obtenue à partir de certaines opérations linéaires appliquées sur ses voisins. On utilise souvent des filtres :

- Les filtres « Passe-haut » qui appliquent une opération de dérivée sur l'image.
- Les filtres « Passe-bas » qui réalisent un lissage

#### 2.1.1.1. Filtres passe-haut

Un filtre « passe haut » est un filtre qui favorise les hautes fréquences, d'où l'appellation de « passe-haut », comme les détails. De ce fait, il améliore le contraste. Un filtre « passe haut » est caractérisé par un masque comportant des valeurs positives et négatives autour du pixel central.

#### 2.1.1.2. Filtres passe-bas :

Les filtres « passe bas » agissent en sens inverse des filtres « passe haut » et le résultat est, un adoucissement des détails, ainsi qu'une réduction du bruit. Le filtre passe-bas se débarrasse des signaux ayant une haute fréquence et qui sont caractérisés par de grandes

variations des niveaux de gris par rapport aux pixels voisins. Il est appelé aussi filtre de lissage car il adoucit considérablement les contours de l'image. Parmi les filtres passe-bas on peut citer les filtres « Moyen » et « Gaussien ».

### *i. Filtre moyen :*

C'est un filtre qui a pour principe de remplacer la valeur de chaque pixel par la moyenne effectuée sur un certain voisinage.

### *ii. Filtre gaussien :*

En traitement d'images, on traite des données à deux dimensions, on donne alors la formule gaussienne à deux dimensions comme suit :

$$G(x, y) = \frac{1}{2\pi\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (1.1)$$

Comme les données traitées dans ce domaine sont discrètes, on est amené à utiliser un masque discret, c'est-à-dire une matrice. Le filtre gaussien a l'avantage de régler le degré de filtrage grâce au paramètre  $\sigma$ .

Les filtres linéaires présentent un inconvénient car ils génèrent un flou au niveau des zones de transitions.

## **2.1.2. Filtres non linéaires**

Parmi les filtres non linéaires nous allons présenter le filtre médian donnant de bons résultats en imagerie.

### **2.1.2.1. Filtre médian**

Le filtre médian est utilisé pour atténuer des pixels isolés, c'est-à-dire ayant une valeur très différente, de leur voisinage. Ce filtre est efficace sur des images dégradées par une source de bruit impulsif, c'est-à-dire une source qui permet à quelques pixels de l'image de prendre des valeurs complètement aléatoires. Le principe de ce filtre est de :

- Considérer une fenêtre centrée sur un pixel ;
- Trier les pixels de cette fenêtre selon un ordre croissant de leur niveau de gris ;

- Affecter au pixel central la valeur médiane.

La valeur médiane est celle qui sépare une séquence triée des valeurs des niveaux de gris en deux parties égales.

Cependant ce filtre possède un inconvénient car il affecte la géométrie des régions de l'image, de ce fait les zones comportant des angles aigus ont tendance à voir leurs angles arrondis par le filtrage.

### 2.1.2.2. Les filtres morphologiques

La morphologie mathématique a comme principe de base de comparer les objets d'une image  $A$  à un certain objet  $B$  pris comme référence (forme géométrique appelée élément structurant) [LAS,04]. Le filtre morphologique est utilisé pour éliminer les pixels isolés considérés comme un bruit. Les opérations de base sont la dilatation et l'érosion, à partir desquelles on peut construire des outils plus avancés, tels que l'ouverture et la fermeture.

Avant de définir les différents opérateurs de la morphologie mathématique, nous tenons d'abord à donner la définition de l'élément structurant.

**Élément structurant** : Un élément structurant est une forme de référence qui peut être quelconque et qui est déplacé sur un ensemble de données [LAS,04].

Les différentes opérations de la morphologie mathématique sont la dilatation, l'érosion, l'ouverture, et la fermeture. Elles sont définies comme suit :

**Dilatation** : Concerne les points noirs isolés au milieu des parties blanches. En pratique, ceci est réalisé en faisant passer sur l'image une fenêtre de taille fixe et en effectuant pour chaque pixel de l'image un OU logique des pixels formant la fenêtre, à l'exception du pixel traité (pixel central).[LAS,04]

**Erosion** : L'érosion est l'opération duale de la dilatation. Les points blancs d'une image sont noyés par les zones noires qui se trouvent autour. Pratiquement, on effectue un ET logique entre les pixels contenus dans la fenêtre utilisée, sauf le pixel central. [LAS,04]

A partir de ces deux opérations de base, d'autres opérations ont été définies :

**Ouverture** : L'ouverture est constituée par une opération d'érosion suivie d'une dilatation.

**Fermeture:** La fermeture est constituée par une opération de dilatation suivie d'une érosion.

## 2.2. Modification d'histogramme

Dans cette approche, on effectue une transformation d'intensité de chaque pixel. Cette transformation est choisie croissante de façon à conserver les liens relatifs entre les régions (une région claire sur un fond sombre apparaîtra plus claire que ce fond dans l'image transformée) [COC, 95].

Les méthodes de modification d'histogramme n'affectent pas la forme des régions, elles en modifient uniquement l'apparence visuelle. [COC, 95].

### 2.2.1. Expansion de la dynamique

L'expansion de la dynamique permet de distribuer les pixels d'une manière uniforme selon une échelle de niveaux de gris. Si une image  $I$  possède des niveaux de gris entre  $G_{\min}$  et  $G_{\max}$  alors, elle deviendra  $I'$  comme suit:

$$I'(x, y) = \frac{I(x, y) - G_{\min}}{G_{\max} - G_{\min}} \quad (1.2)$$

### 2.2.2. Egalisation d'histogramme

Elle a pour but de rendre le plus plat possible l'histogramme des niveaux de gris dans l'image. Le niveau de gris de chaque pixel est calculé comme suit: [COC, 95]

$$G' = \frac{255}{\text{nombre\_pixel\_total}} \text{Histocumulé}(G) \quad (1.3)$$

Histocumulé ( $G$ ) : le nombre total de pixels ayant le niveau de gris inférieur ou égal à  $G$

## 2.3. Rehaussement du contraste

Le rehaussement de contraste a pour but de supprimer l'effet de flou, qui est caractérisé par une transition s'étalant sur plusieurs pixels entre deux régions différentes, dû par exemple à un bougé lors de la prise de vue ou de défocalisation de la caméra. Le principe consiste à essayer de restaurer les contours et de diminuer l'étendue de la zone de transition

sans toucher aux zones homogènes de l'image. Ainsi, on limite le risque de fusionner deux régions différentes.

### 3. Segmentation

#### 3.1. Introduction

L'homme sait naturellement séparer des objets dans une image. Pour séparer un objet d'un fond, l'humain se base sur des connaissances de haut niveau qui lui permettent de détecter dans l'image ce qui l'intéresse. Le but de la segmentation d'image est aussi d'être capable de séparer différentes zones homogènes significatives de l'image et qui serviront de base pour d'éventuels traitements de haut niveau comme : la reconnaissance de formes.

Il existe plusieurs techniques de segmentation. En effet, chacune d'entre elles, possède des avantages et des inconvénients et chaque technique est plus ou moins adaptée à un certain type d'image.

#### 3.2. Définition formelle de la segmentation

Formellement, la segmentation d'une image A en régions  $R_i$ ,  $i = 1..n$ , est définie par les propriétés suivantes :

$$A = \bigcup_i^n R_i$$

$$\forall i, j ; i \neq j \quad R_i \cap R_j = \phi \quad (1.4)$$

$$\forall i, i = \overline{1, n} \quad P(R_i) = \text{vrai}$$

$$\forall i, j \quad i = \overline{1, n}, j = \overline{1, n} \quad P(R_i \cup R_j) = \text{faux}, R_i \text{ et } R_j \text{ sont deux régions voisines.}$$

P est un prédicat d'homogénéité.

La première condition indique que l'union des régions nous ramène à l'image de départ. Ce qui signifie que la segmentation est complète.

La deuxième indique que deux régions différentes sont disjointes.

La troisième exprime que les pixels appartenant à une région doivent satisfaire le critère d'homogénéité.

La quatrième exprime le critère d'homogénéité pour la segmentation en régions disjointes.

### 3.3. Les approches de la segmentation

La segmentation fait référence aux notions de différence et de similarité des pixels, ce qui donne naissance à deux approches duales :

- Approche contours (recherche des discontinuités dans l'image).
- Approche région (recherche des zones de niveaux de gris homogènes)

#### 3.3.1. Approche contours (frontières)

Dans cette approche, on s'intéresse aux variations d'intensités des pixels, ces variations constituent les frontières des régions correspondant à des bords ou à des parties d'objets de la scène. Plusieurs méthodes ont été développées pour ce type d'approche afin d'arriver à une détection qui fournit une représentation correcte de l'image et donner toutes les propriétés significatives de la scène.

##### 3.3.1.1. Méthodes dérivatives

Pour déterminer la position des éventuels contours, on utilise les méthodes dérivatives. L'image dérivée met en évidence les variations de niveaux de gris ; quand il y a une forte variation, en général c'est dû à un changement de zone.

En imagerie à deux dimensions une scène est modélisée par une fonction de niveau de gris  $f(x,y)$  et les contours sont assimilés aux points de fort gradient (par la recherche des maximums de la dérivée première) ou au passage par zéro de la dérivée seconde.

Deux techniques ont été définies pour cette méthode :

##### *i. Utilisation du gradient*

La détection d'un contour se fait par la recherche des maximums du gradient. Soit  $G$  la fonction de niveau de gris de l'image à traiter. Le gradient est donné par :

$$\nabla G(x, y) = \left( \frac{\partial G(x, y)}{\partial x}, \frac{\partial G(x, y)}{\partial y} \right) \quad (1.5)$$

En général, on utilise sa norme au lieu d'utiliser ses coordonnées directement pour pouvoir la comparer à un seuil et déterminer où sont les contours. La norme du gradient s'exprime par :

$$|\nabla G(x, y)| = \sqrt{\left(\frac{\delta G(x, y)}{\delta x}\right)^2 + \left(\frac{\delta G(x, y)}{\delta y}\right)^2} \quad (1.6)$$

Ou :

$$|\nabla G(x, y)| = \max\left\{\left|\frac{\delta G(x, y)}{\delta x}\right|, \left|\frac{\delta G(x, y)}{\delta y}\right|\right\} \quad (1.7)$$

Ou encore : [LAS,04]

$$|\nabla G(x, y)| = \frac{\left\{\left|\frac{\delta G(x, y)}{\delta x}\right| + \left|\frac{\delta G(x, y)}{\delta y}\right|\right\}}{2} \quad (1.8)$$

Tandis que l'orientation du gradient est donnée par :

$$\varphi(x, y) = \arctan\left[\frac{\delta G(x, y)}{\delta y} / \frac{\delta G(x, y)}{\delta x}\right] \quad (1.9)$$

En pratique, on dispose de masques numériques à balayer sur l'image. Les plus connus sont les opérateurs de Prewitt, ceux de Sobel et Robert.

### *ii. Utilisation du Laplacien*

Une deuxième méthode pour détecter les contours est l'utilisation du Laplacien. Le but de ce dernier est la recherche des passages par zéro de la dérivée seconde. Son principe de calcul est donné par la formule suivante:

$$\Delta G(x, y) = \frac{\delta^2 G(x, y)}{\delta x^2} + \frac{\delta^2 G(x, y)}{\delta y^2} \quad (1.10)$$

Le Laplacien est utilisé sous sa forme discrète qui peut être estimée par une convolution avec des masques 3x3 ou 5x5.

Pour les méthodes dérivatives et afin d'accentuer les zones de haute intensité de gradients, le calcul de ces derniers est généralement suivi par un seuillage. Le seuillage élimine les transitions de faible gradient pouvant correspondre à du bruit.

L'inconvénient des méthodes dérivatives est qu'elles sont très sensibles au bruit, et l'image obtenue comporte beaucoup de contours, même ceux qui ne sont pas significatifs dans l'image.

### 3.3.1.2. Approche par filtrage optimal

#### *i. Problématique du contour*

Un des problèmes fondamentaux de la détection des contours est dans la résistance au bruit présent dans l'image (bruit du capteur, bruit d'échantillonnage, irrégularités de la surface des objets). On est donc confronté au problème de la différentiation d'un signal bruité ; pour y remédier il est d'abord nécessaire de :

- Définir des critères de performance.
- Dédire de ces critères une famille de filtres optimaux.
- Résoudre le problème.

Un détecteur de contour doit satisfaire les conditions suivantes :

- Les contours de l'image ne doivent pas être perdus ;
- Pas de génération de faux contours ;
- Les points de contours doivent être bien localisés ;
- Eliminer la possibilité de réponses multiples à un contour.

#### *ii. Critères de Performance [COC,95]*

La réponse est choisie de telle sorte à satisfaire les 3 critères

- Une bonne détection,
- Une bonne localisation,
- Une faible multiplicité des maximums dus au bruit.

Plusieurs méthodes de calcul du filtre optimal pour détecter et localiser les contours sont proposées.

### iii. Filtre de Canny

On a vu que les effets de bruit pouvaient être atténués par convolution de l'image avec un filtre passe-bas. Il est possible d'effectuer ce filtrage puis de dériver l'image pour diminuer l'influence du bruit, mais il est aussi possible de dériver directement le produit de convolution de l'image et du filtre. Nous prendrons l'opérateur de Canny qui dérive la convolution d'une image avec une Gaussienne.

La méthode suivie par Canny est comme suit :

Soit la Gaussienne :

$$G(x, y, \sigma) = (1/\sqrt{2\pi}\sigma) e^{-(x^2 + y^2)/2\sigma^2} \quad (1.11)$$

La dérivée de l'image  $I(x, y)$  filtrée est :

$$I_f(x, y) = \nabla(G(x, y) \otimes I_i(x, y)) \quad (1.12)$$

Avec :

$I_i(x, y)$  : Image d'entrée (initiale)

$G(x, y)$  : Fonction de Gauss

$I_f(x, y)$  : Image de sortie (finale)

$$I_f(x, y) = (\partial/\partial x + \partial/\partial y)(G(x, y) \otimes I_i(x, y)) = G'_x(x, y) \otimes I_i(x, y) + G'_y(x, y) \otimes I_i(x, y)$$

$$I_f(x, y) = fx + fy$$

Avec :

$$G'_x(x, y) = (-x/\sqrt{2\pi}\sigma^3) e^{-(x^2+y^2)/2\sigma^2}, \quad G'_y(x, y) = (-y/\sqrt{2\pi}\sigma^3) e^{-(x^2+y^2)/2\sigma^2}$$

Les filtres étant séparables, nous pouvons réaliser séparément les convolutions suivant  $x$  et  $y$ :

$$G'_x(x, y) = G'_x(x) \otimes G(y) \quad \text{et} \quad G'_y(x, y) = G'_y(y) \otimes G(x)$$

$$D'où \quad fx = G'_x(x, y) \otimes I_i(x, y) = G'_x(x) \otimes G(y) \otimes I_i(x, y)$$

$$Et \quad fy = G'_y(x, y) \otimes I_i(x, y) = G'_y(y) \otimes G(x) \otimes I_i(x, y)$$

La direction et l'amplitude sont alors données par les relations suivantes :

$$A = \sqrt{f_x^2 + f_y^2} \quad \text{et} \quad \text{Dir} = \text{Actan}(f_y/f_x).$$

#### **iv. Filtre de Deriche**

Deriche, utilisant la même démarche que Canny, a cherché une réalisation de l'opérateur sous la forme d'un filtre à réponse impulsionnelle infinie [COC, 95]. Pour améliorer l'immunité au bruit un lissage au préalable est effectué. Le principe consiste à effectuer un lissage qui est la combinaison de deux filtres monodimensionnels dans les directions x et y.

Le calcul du gradient se fait à partir des dérivées selon x et y du produit de convolution de l'image par le filtre de lissage  $f(x,y)$ .

L'avantage de ce filtre, est qu'il peut être implémenté de manière récursive et qu'il nécessite très peu d'opérations par pixel. [DER, 95]

### **3.3.2. Approche région**

Elle fait référence à des groupements de points ayant des propriétés communes. Les méthodes de segmentation par régions aboutissent directement à une partition de l'image ; chaque pixel étant affecté à une région unique vérifiant un certain nombre de propriétés [CHR,03] :

- l'homogénéité intérieure
- l'hétérogénéité avec les zones voisines

Différentes méthodes de segmentation par régions sont définies. Nous citons dans ce qui suit quelques unes.

#### **3.3.2.1. Segmentation par fusion (merge)**

L'initialisation de cette méthode consiste à considérer chaque pixel comme une région. On fusionne les régions entre elles avec un double critère de similarité des niveaux de gris et d'adjacence de régions. On s'arrête quand le critère de fusion n'est plus vérifié.

#### **3.3.2.2. Segmentation par division (Split)**

Cette méthode suppose au départ que tous les pixels appartiennent à la même région ; si celle-ci est homogène, on s'arrête sinon elle est divisée en plusieurs régions. Le

processus est répété pour chaque région nouvellement créée. On peut ajouter, lors de la division, un critère d'arrêt pouvant être la taille des régions.

### 3.3.2.3. Segmentation par division-fusion (split and merge)

Les segmentations par division-fusion [HOR,74] ont pour principe d'effectuer une découpe arbitraire de l'image, de réaliser des tests de "cohésion" des zones puis de modifier ces zones.

Le fonctionnement de l'algorithme de division-fusion (split & merge) est le suivant : Dans un premier temps, on commence par découper l'image en zones de taille moyenne. On parcourt chacune des régions en cherchant si elle vérifie un critère d'homogénéité (exemple : variance, texture...). Si la zone est homogène, on ne fait rien, sinon, on la divise en quatre sous régions jusqu'à avoir des régions homogènes. On commence alors l'étape de fusion pendant laquelle on regroupe les zones contiguës correspondant à la même région. On obtient ainsi une segmentation de l'image mais sans avoir de frontières précises entre les objets.

En fait, un grand nombre de variantes sont possibles en fonction de la taille minimale des zones utilisés et du critère d'homogénéité.

### 3.3.2.4. Techniques de seuillage

Le seuillage est considéré comme étant la plus simple technique de classification. Les techniques de classification sont celles qui ont pour but d'organiser un ensemble de données en classes [LAS, 04]. Dans le cas de la classification par binarisation, les pixels de l'image sont partagés par un seul seuil  $s$  en deux classes fond et objet. Le problème de la binarisation est le choix du seuil ; sachant qu'à partir d'une image en niveaux de gris, nous pouvons obtenir plusieurs images binaires, il est alors difficile de dire laquelle est correcte.

Dans des cas simples, la moyenne ou la valeur médiane des pixels de l'image peut jouer le rôle d'un seuil. Aussi, si l'histogramme est bimodal, nous choisissons alors le seuil comme étant le point le plus inférieur entre les deux pics. [LAS, 04].

Les techniques de seuillage sont malheureusement très sensibles aux bruits et ne peuvent être employées que dans des applications spécifiques.

### 3.3.3. Autres approches

#### 3.3.3.1. Les modèles actifs

Ce type d'approche assez différent des précédents a été introduit dans [KAS, 87] sous le nom de « *Snakes* ». Un « *Snake* », qui se traduit par serpent, est une sorte de ballon gonflable qui doit venir épouser le contour de l'objet que l'on cherche. Dans cette méthode, on utilise une courbe qui va se déformer sous l'influence d'une énergie interne que l'on fournit au « *Snake* » pour renforcer sa déformation et d'une énergie externe (provenant de l'image) pour s'approcher au maximum du contour. On initialise une forme (généralement un cercle ou une ellipse) que l'on va "gonfler" ou "dégonfler" jusqu'à ce qu'elle épouse au mieux le contour de l'objet à détecter.

#### 3.3.3.2. Recherche de patrons

Dans cette approche, on utilise une forme géométrique appelée aussi patron (template) que l'on va tenter de localiser sur l'image.

Le principe est de connaître exactement (ou presque) la forme de l'objet que l'on recherche et de parcourir toute l'image pour placer le template à l'endroit le plus vraisemblable. Ceci est beaucoup plus contraignant que d'avoir besoin de savoir combien d'éléments sont présents dans l'image. Il faut, dans ce cas connaître la forme précise de celle que l'on recherche. Une forte similarité indique une forte probabilité d'avoir trouvé le motif.

Cette méthode a été appliquée à une recherche de visage dans une séquence vidéo.

## 4. Conclusion

Nous avons présenté dans ce chapitre les différentes méthodes et techniques de prétraitement et de segmentation d'images; cette dernière est loin d'être un problème résolu.

Les méthodes de segmentation étudiées présentent quelques inconvénients. L'approche contour, par exemple nécessite souvent un algorithme de fermeture du contour. L'approche région est, quant à elle généralement plus gourmande en temps calcul. Actuellement, de nouvelles méthodes sont apparues dans les deux types d'approches. Dans ce qui suit nous nous intéresserons plus en détail à la segmentation par contours actifs qui permettent d'améliorer la qualité de la segmentation.

## Chapitre 2

### Contours Actifs et les algorithmes génétiques

#### 1. Introduction

La segmentation d'images constitue la première étape d'un grand nombre d'applications dans le traitement des images, elle consiste à partitionner l'image en zones homogènes. Il existe plusieurs façons d'aborder le problème en pratique :

- Par des méthodes directes, qui font intervenir des opérateurs de segmentation classiques sur l'image (par exemple : les opérateurs gradient) ;
- Par des méthodes qui manipulent une forme englobant l'objet à extraire et qui la font évoluer d'une position initiale vers une position finale correspondant à l'objet ciblé.

Cette dernière, c'est la méthode introduite récemment par [KAS, 88], est présentée dans ce présent chapitre. Elle est celle des contours actifs. Cette méthode a pour but de faire évoluer une courbe élastique dans l'image à segmenter afin de l'amener à converger vers les zones de fortes gradients.

#### 2. 2. Historique des modèles déformables:

L'étude des formes et de leurs déformations au cours du mouvement constitue un thème important en vision par ordinateur, dans ce cadre les chercheurs se sont beaucoup intéressés aux modèles d'objets 2D et 3D.

Selon le principe proposé par Kass, Witkin et Terzopoulos [KAS,88], les modèles déformables sont basés sur les lois de la physique et de la dynamique. Plus précisément l'évolution des modèles déformables est en général gérée par l'application du principe de moindre action de Hamilton. Selon ce principe, l'évolution réelle d'un système physique entre un état initial ( $t_0$ ) et un état final ( $t_1$ ) donnés est un point stationnaire pour l'action

$$A = \int_{t_0}^{t_1} (\text{énergie cinétique} - \text{énergie potentielle}) dt$$

Les modèles déformables dits « physiques » utilisent une représentation déterministe et physique de la courbe (par des points, des ressorts,...) et prêtent à la courbe des propriétés physiques telles que l'élasticité. Kass et Al [KAS, 88] ont introduit en vision par ordinateur la notion de contours

actifs ou « snakes ». Ce sont des modèles élastiques des courbes en mouvement, que l'on peut coupler à des éléments particuliers des images, tels que les contours.

L'optimisation des *snakes* s'effectue par minimisation d'énergie. Par un choix judicieux de cette énergie, les *snakes* sont ainsi guidés vers les primitives aux quelles on s'intéresse au cours de la minimisation de l'énergie [MEN, 90].

Zucker et al [ZUC ,88],[DAV,89] ont développé pour l'analyse d'empreintes digitales des *snakes* particuliers pour lesquels un contour courbe est décrit par un ensemble de splines de longueurs assez faibles qui se chevauchent 2 à 2 et dont l'union couvre tout le contour.

La minimisation de l'énergie du modèle déformable atteint donc en général un minimum local. Cette méthode nécessite par conséquent la donnée d'une initialisation appropriée pour converger vers une solution significative.

Cohen [Coh 89], [Coh91] a proposé un nouveau type de contours actifs, les ballons, qui sont beaucoup moins sensibles à l'initialisation que les contours actifs classiques.

En effet, l'initialisation, qui est supposée interne au contour, peut être très éloignée de lui. Pour traiter ce cas, Cohen [Coh ,89], [Coh,91] préconise l'utilisation d'une force de dilatation normale à la courbe qui agit comme une pompe dans un ballon afin que le « *snake* » puisse atteindre le contour recherché.

Les modèles déformables physiques de types « *snakes* » ont donc été introduits en vision dans le contexte de la segmentation statique d'images et sont largement utilisés dans ce contexte.

Néanmoins, ces modèles ont aussi été appliqués à d'autres tâches de la vision, telles que l'extraction de régions, la restauration d'images, le suivi de structures déformables sur des séquences d'images.

Les algorithmes de contours actifs sont de plus en plus utilisés dans les domaines de l'imagerie médicale et de la biotechnologie pour, par exemple, identifier le contour d'une tumeur lors d'un examen échographique ou dermatologique, ou pour faire du tri cellulaire lors d'un examen cytologique.

Les avantages qu'offre l'approche permettent d'élargir les applications à des problèmes liés à d'autres modalités d'images, exemple la détection automatique des lignes côtières en imagerie satellites radars ou la segmentation de la pupille dans les images de la rétine.

### 2.1. Qu'est ce qu'un contour actif ?

Les contours actifs ou « *snakes* » sont des courbes fermées ou ouvertes et à extrémités fixes ou non. Ils ont une position initiale qui est près de l'objet à extraire ou à distinguer du reste de l'image. Ils se déforment et se déplacent dans l'image à cause des forces auxquelles ils sont soumis [ROU,

03].

Les contours actifs peuvent être divisés en trois catégories :

- les contours actifs fermés (le point de départ et le point d'arrivée sont les mêmes) ;
- les contours actifs à extrémités libres ;
- les contours actifs à extrémités fixes où les positions du point de départ et du point d'arrivée sont fixes dans le temps ainsi que les dérivées premières en ces points (les points de départ et d'arrivée restent fixes).

La figure 2.1 montre des exemples de chaque type de contours actifs

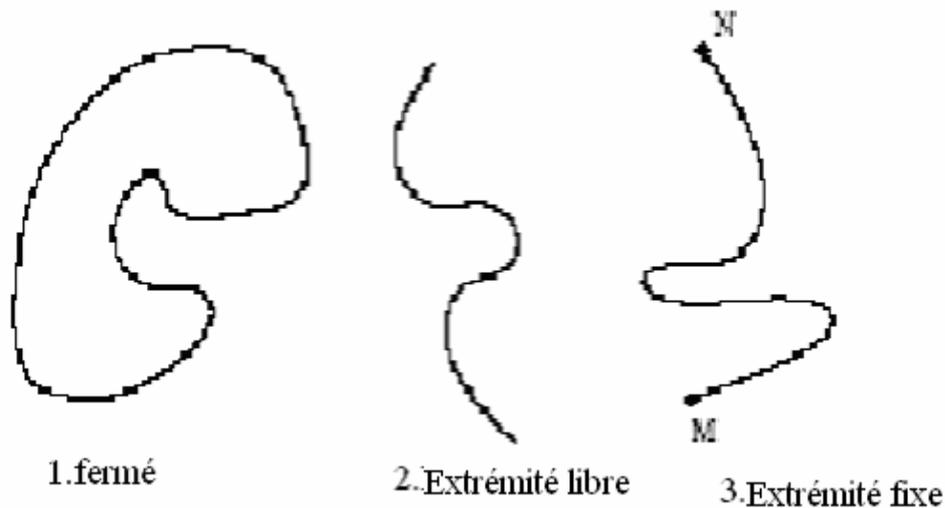


Figure 2.1: Types de contours actifs

## 2.2. Principe des contours actifs

Le principe des « *snakes* » est de faire évoluer un contour fermé initial autour de l'objet d'intérêt vers une position d'équilibre, c'est à dire en direction des bords de l'objet à détecter. Entre chaque deux positions différentes du contour (deux itérations), le mouvement des points est fait par une équation impliquant des forces qui agissent sur le contour. Ces forces dépendent des données de l'image (tel que le gradient et l'intensité) et des propriétés du contour (rigidité, élasticité). La position d'équilibre du contour peut être prise comme étant le minimum d'une fonction d'énergie [KHE, 01].

La figure 2.2 illustre le principe du contour actif :

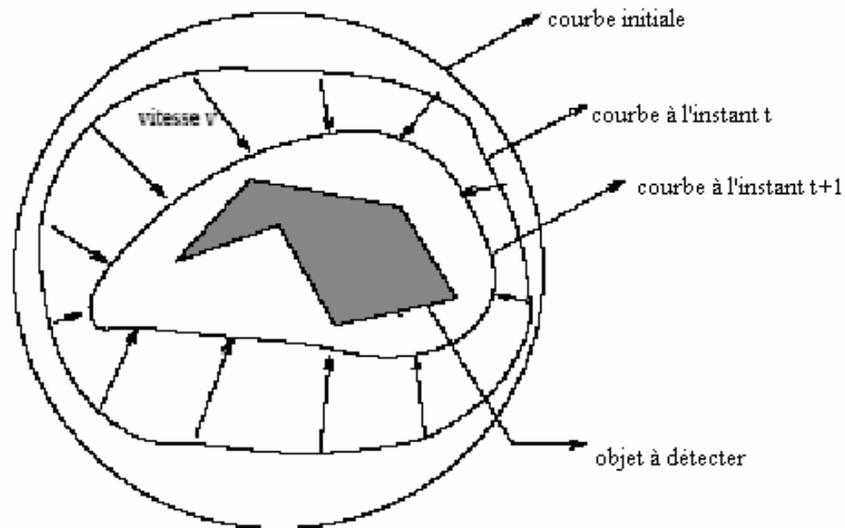


Figure 2.2: Principe des contours actifs.

Le modèle des contours actifs « *snakes* » consiste en l'initialisation d'une courbe (fermée) à proximité des frontières de l'objet à détecter. Cette courbe notée  $C$  se déplace selon un processus itératif de déformation contrôlée par un test de convergence. Le processus de déformation est lié à la minimisation d'une fonction d'énergie, construite de telle sorte qu'un minimum local se trouve en frontière avec l'objet à détecter.

Le contour est défini par une représentation paramétrique  $v$ ,

$$\Omega : [0,1] \longrightarrow \mathbb{R}^2$$

$$s \longrightarrow v(s) = [x(s), y(s)]$$

où  $s$  est l'abscisse curviligne ;

$v(s)$  est le vecteur de position du point de contour de coordonnées  $x(s)$  et  $y(s)$  ;

$v(1)$  et  $v(0)$  sont les vecteurs de position des extrémités du contour ;

$v(1) = v(0)$  pour un contour fermé.

### 3. Les énergies

La fonction d'énergie relative au contour actif est composée de trois énergies

$$E(v) : v \rightarrow E_{\text{int\`eme}}(v) + E_{\text{externe}}(v) + E_{\text{contexte}}(v) \quad (2.1)$$

On va préciser ces trois types d'énergies.

#### 3.1 L'énergie interne

L'énergie interne gère la cohérence de la courbe. Elle maintient la cohésion des points et la raideur de la courbe qui est utilisée comme terme de régularisation.

$$E_{\text{int\`eme}}(v) = \int_0^1 \left( \frac{\alpha}{2}(s) \|v'(s)\|^2 + \frac{\beta}{2}(s) \|v''(s)\|^2 \right) ds \quad (2.2)$$

Les termes  $v'$  et  $v''$  sont les dérivées première et seconde de  $v$  par rapport à  $s$ , « le terme du 1<sup>er</sup> ordre » correspond à la tension. Il prend une valeur importante quand la courbe se distend. Lorsque  $\alpha = 0$  la courbe peut présenter des discontinuités. On parlera donc d'énergie de continuité.

« Le terme du 2<sup>ème</sup> ordre » correspond à la courbure. Il prend une valeur importante, lorsque la courbe s'incurve rapidement c'est-à-dire pour l'obtention de coins. Lorsque  $\beta = 0$ , la courbe peut prendre une forte convexité, par contre lorsque  $\beta$  est grand, la courbe tendra vers un cercle si elle est fermée ou une droite si elle est ouverte.

#### 3.2. Energie externe

L'énergie interne qu'on vient d'analyser, gère la régularisation du contour actif. L'énergie externe correspond à l'adéquation aux données. Cette énergie externe prend en compte les caractéristiques de l'image. On rappelle ici que ce sont les contours de formes qui sont recherchés donc les points de fort gradient ou des points ayant une propriété de position par rapport à une couleur donnée.

##### 3.2.1 Gradient

Pour la recherche des zones de fort contraste dans l'image, est introduite la fonction [ANW,01] :

$$E_{\text{externe}}(v) = -\int_0^1 \|\nabla I(v(s))\|^2 ds \quad (2.3)$$

où  $\nabla I(v(s))$  représente le gradient de l'image  $I$  en  $v(s)$ .

### 3.2.2 Intensité :

Cette énergie, au contraire, permet de sélectionner les zones sombres ou claires selon le signe choisi.

$$E_{\text{intensité}} = \pm \int_0^1 (I(v(s)) - i_0)^2 ds \quad (2.4)$$

La valeur  $i_0$  introduit ou non, un certain seuillage. On peut ainsi favoriser la position du contour dans une zone donnée.

### 3.3. Energie de contexte

L'énergie de contexte, parfois appelée énergie de contrainte, permet d'introduire des connaissances a priori sur ce que l'on cherche. Entre autres, on place, sous cette rubrique, l'énergie ballon introduite par Cohen [COH,91]. La force ballon va gonfler le contour actif ou accélérer sa rétraction selon le signe de la force introduite. De plus, cette force va permettre de dépasser les contours présentant un faible gradient et ainsi de sortir du bruit pour atteindre une frontière plus fortement marquée. Il s'agit d'une force normale au contour en chaque point.

$$\vec{F}_{\text{ballon}}(V(s)) = K\vec{n}(s) \quad (2.5)$$

Où  $\vec{n}(s)$  est un vecteur unitaire normal à la courbe en  $v(s)$ .

## 4. Méthodes d'optimisation

Plusieurs méthodes ont été proposées pour résoudre ce problème. Trois principales familles d'approches sont généralement utilisées dans les applications décrites dans la littérature :

- la famille des approches variationnelles [KAS,87] qui tire avantage des

développements mathématiques de l'analyse numérique ;

- la famille des approches reposant sur la programmation dynamique [AMI,88] qui utilise les avancées de l'informatique ;
- la famille des approches autour de l'algorithme de Greedy [SHA, 92] qui est appréciée pour son temps de traitement plus rapide [DEN, 95].

On se place dans la troisième approche, dans le cadre d'une implémentation reposant sur un algorithme de Greedy

#### 4.1 Approche variationnelle [COU,05]

Dans sa forme continue, le « *snake* » est modélisé par une courbe paramétrée :

$$V(s) = (x(s); y(s))^t, s \in [a, b] \quad (2.6)$$

où  $s$  est généralement l'abscisse curviligne.

Le contour initial  $v_0$  est défini par l'utilisateur et la courbe évolue avec une certaine vitesse (figure 2.2). Le problème est de trouver cette vitesse telle que la courbe évolue vers un minimum local correspondant aux contours des objets ou régions à segmenter.

Suivant les considérations précédentes concernant les différentes énergies mises en jeu, l'énergie totale du « *snake* » est donc donnée par :

$$E_{snake} = \int_b^a [E_{interne}(v(s)) + E_{image}(v(s)) + E_{externe}(v(s))] ds \quad (2.7)$$

L'énergie interne dépend uniquement de la forme du *snake*. C'est une contrainte de régularité de la courbe (elle produira une courbe régulière). Cette énergie s'écrit :

$$E_{interne} = \varpi_1(s).E_{élasticité} + \varpi_2(s).E_{courbure} = \varpi_1(s).\left(\frac{dv}{ds}\right)^2 + \varpi_2(s).\left(\frac{d^2v}{ds^2}\right)^2 \quad (2.8)$$

Où  $\varpi_1$  est le facteur d'élasticité et  $\varpi_2$  le facteur de rigidité du contour. Le premier coefficient contrôle le terme de premier ordre qui agit sur la continuité et le deuxième contrôle le terme du second ordre qui agit sur la courbure.

L'énergie potentielle liée à l'image est donnée par :

$$E_{image} = -\lambda(s)|\nabla I(v(s))|^2 \quad (2.9)$$

Où le facteur  $\lambda$  dépend de l'image  $I$  initiale. Cette énergie caractérise les éléments vers lesquels on veut attirer le *snake* sur l'image. Le choix du gradient de l'image au point considéré est effectué car pour la détection de contours, on s'intéresse aux lignes de fort gradient.

L'énergie externe (ou de contraintes) est définie par l'utilisateur selon les spécificités du problème. Par exemple, imposer une distance minimale ou maximale entre deux points

consécutifs du contour actif.

En l'absence de  $E_{\text{externe}}$ , et dans les nombreux cas où l'on recherche les lignes de fort gradient, on peut alors écrire :

$$E_{\text{snake}} = \int_b^a \left[ -\lambda(s) |\nabla I(v(s))|^2 + \varpi_1(s) \left( \frac{dv}{ds} \right)^2 + \varpi_2(s) \left( \frac{d^2s}{ds^2} \right)^2 \right] ds \quad (2.10)$$

Une fois les énergies de chaque point du voisinage calculées, on décidera vers quel pixel de l'image le point du *snake* va migrer (vers le point du voisinage dont la somme des trois énergies est la plus faible).

Le premier terme est un terme d'attache aux données permettant d'attirer la courbe vers les forts gradients de l'image tandis que les deux derniers termes sont utilisés pour imposer une contrainte de régularité sur le contour. Le fait de prendre

$\beta = 0$  autorise les discontinuités du second ordre tels les angles aigus d'un objet.

#### 4.1.1 Mise en équation

On désigne par  $v'$  et  $v''$  les dérivées premières et secondes de  $v$  le long de la courbe et par  $\Omega$  la région, l'énergie à minimiser est donc donnée par :

$$E(v(s)) = \int_{\Omega} \varpi_1(s) (v'(s))^2 ds + \int_{\Omega} \varpi_2(s) (v''(s))^2 ds - \int_{\Omega} \lambda(s) |\nabla I(v(s))|^2 ds \quad (2.11)$$

Où  $\varpi_1$ ,  $\varpi_2$  et  $\lambda$  sont potentiellement variables le long de  $s$ . Cette équation intégrale est résolue généralement de façon variationnelle. On suppose que le contour évolue vers un minimum d'énergie.

Le minimum de l'énergie est atteint lorsque l'équation associée d'Euler-Lagrange est satisfaite :

$$(\varpi_1 v')' + (\varpi_2 v'')'' + \nabla P(v) = 0 \quad (2.12)$$

où

$$P(v) = -\lambda(s) |\nabla I(v(s))|^2 \quad (2.13)$$

C'est le potentiel associé aux forces externes. On obtient finalement :

$$-\varpi_1 v''(s) + \varpi_2 v^{(4)}(s) + \nabla P(s) = 0 \quad (2.14)$$

En supposant  $\varpi_1$ ,  $\varpi_2$  et  $\lambda$  des constantes positives.

#### 4.1.2 Discrétisation et résolution

Il y a deux façons de concevoir la discrétisation d'une courbe :

**1. selon les différences finies** : les éléments de la courbe sont réduits en des points auxquels sont attachés les éléments mécaniques (masse, raideur.. .) de la courbe considérée en ces points ; cette méthode consiste à remplacer les dérivées apparaissant dans le problème continu par des différences finies.

**2. selon les éléments finis** : on remplace chaque portion de courbe par le segment élémentaire et les éléments mécaniques sont calculés sur ces segments, cette méthode consiste à approcher, dans un sous-espace de dimension finie, un problème écrit sous forme variationnelle (minimisation de l'énergie en général).

Nous ne décrivons ici que la première approche la plus fréquemment employée.

Nous discrétisons alors la courbe du *snake* en un nombre  $n$  de points, alors nous pouvons poser :

$$V^t = [v_0, v_1, v_2, \dots, v_{n-1}]^t$$

Afin de simplifier l'écriture, nous allons écrire pour la somme des forces liées à l'image  $F(v) = -\nabla P(v)$ . Ainsi, en reprenant l'équation de l'énergie à minimiser (avec  $\omega_1$  et  $\omega_2$  constants), nous pouvons écrire :

$$-\omega_1 v''(s) + \omega_2 v^{(4)}(s) = F(v(s)) \quad (2.15)$$

Or, en écrivant  $v_i$  la solution approchée de la solution exacte  $v(ih)$  pour  $i = 0..n-1$  au point  $ih$ , nous pouvons approcher le système par

$$\sum_{i=0}^{n-1} -\omega_1 v''(ih) + \omega_2 v^{(4)}(ih) - F(v(ih)) = 0 \quad (2.16)$$

soit

$$-\omega_1 v_i'' + \omega_2 v_i^{(4)} = F(v_i) \quad (2.17)$$

En utilisant les différences finies par pas  $h$  dans l'espace de (distance entre deux points de l'échantillonnage), nous approchons les dérivées par leur forme centrée (meilleure formule que les formes progressives ou régressives) :

$$\begin{aligned} v_i' &= \frac{v_{i+1} - v_i}{h} + o(h) \\ v_i'' &= \frac{v_{i+1}' - v_i'}{h} = \frac{v_{i+2} - 2v_{i+1} + v_i}{h^2} + o(h^2) \\ v_i''' &= \frac{v_{i+2}' - 2v_{i+1}' + v_i'}{h^2} = \frac{v_{i+3} - 3v_{i+2} + 3v_{i+1} - v_i}{h^3} + o(h^3) \\ v_i^{(4)} &= \frac{v_{i+3}' - 3v_{i+2}' + 3v_{i+1}' - v_i'}{h^3} = \frac{v_{i+4} - 4v_{i+3} + 6v_{i+2} - 4v_{i+1} + v_i}{h^4} + o(h^4) \end{aligned} \quad (2.18)$$

Pour les besoins du problème, nous utiliserons les deux approximations suivantes :

$$v_i'' \approx \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2}$$

$$v_i^{(4)} \approx \frac{v_{i+2} - 4v_{i+1} + 6v_i - 4v_{i-1} + v_{i-2}}{h^4}$$

Maintenant, on peut remplacer dans l'équation de l'énergie :

$$-\frac{\omega_1}{h^2}(v_{i-1} - 2v_i + v_{i+1}) + \frac{\omega_2}{h^4}(v_{i-2} - 4v_{i-1} + 6v_i - 4v_{i+1} + v_{i+2}) = F(v_i) \quad (2.19)$$

$$\text{soit } \frac{\omega_2}{h^4}v_{i-2} + \left(\frac{\omega_1}{h^2} - 4\frac{\omega_2}{h^4}\right)v_{i-1} + \left(2\frac{\omega_1}{h^2} + 6\frac{\omega_2}{h^4}\right)v_i + \left(-\frac{\omega_1}{h^2} - 4\frac{\omega_2}{h^4}\right)v_{i+1} + \frac{\omega_2}{h^4}v_{i+2} = F(v_i)$$

On peut réécrire cette équation sous sa forme matricielle

$$AV = F \quad (2.20)$$

Où A une matrice symétrique car la courbe est fermée, est une matrice pentadiagonale de taille

$n \times n$  fonction de  $w_1$  et  $w_2$ . En posant  $\alpha = \frac{w_2}{h^4}$ ,  $\beta = -\frac{w_1}{h^2} - 4\frac{w_2}{h^4}$  et  $\gamma = 2\frac{w_1}{h^2} + 6\frac{w_2}{h^4}$ .

la matrice A s'écrit alors :

$$A = \begin{pmatrix} \gamma & \beta & \alpha & 0 & \dots & & & & & & \\ \beta & \gamma & \beta & \alpha & 0 & \dots & & & & & \\ \alpha & \beta & \gamma & \beta & \alpha & 0 & \dots & & & & \\ 0 & \alpha & \beta & \gamma & \beta & \alpha & 0 & \dots & & & \\ \vdots & 0 & \alpha & \beta & \gamma & \beta & \alpha & 0 & \dots & & \\ & \vdots & & \ddots & \\ & & & & \ddots \end{pmatrix}$$

En regardant bien la discrétisation (maillage) que l'on a effectuée sur la courbe du « snake », on voit bien que chaque point est lié à ses quatre voisins ce qui explique la matrice obtenue.

Lors de la résolution du problème, nous retrouvons une solution de celui-ci en résolvant l'équation associée de l'évolution avec des conditions aux bords fixées ou périodiques :

$$\frac{\delta v}{\delta t} - (w_1 v') + (w_2 v'')'' = F(v) \quad (2.21)$$

Cette équation revient à effectuer une descente d'énergie vers un minimum.

Ce schéma, associé à une équation d'évolution, aboutit à l'équation (on admettra le résultat) :

$$(I_d + \tau A)V^t = V^{t-1} + \tau F(V^{t-1}) \quad (2.22)$$

$$\text{soit } V^t = (Id + \tau A)^{-1}(V^{t-1} + \tau F(V^{t-1}))$$

Où  $I_d$  est la matrice identité de taille  $n \times n$  et  $\tau$  le pas du temps. Nous avons obtenu alors un système penta diagonal, linéaire symétrique et positif.

Afin de résoudre ce système, on calcule une seule fois la matrice  $(I_d + \tau A)^{-1}$  car  $\omega_1$  et  $\omega_2$  sont invariants le long de la courbe. Dans le cas où ces deux termes ne sont pas constants, il faudrait inverser la matrice à chaque instant  $t$  de pas  $\tau$ . Nous arrêtons d'itérer lorsque la différence entre deux itérations est très petite.

#### 4.2 Algorithme de Greedy

L'utilisation de l'algorithme de Greedy pour minimiser l'énergie totale d'un contour actif, comme l'ont proposé Williams et Shah [SHA, 92] est devenue une alternative assez fréquente à l'approche variationnelle. Williams et Shah, de manière classique discrétisent l'expression :

$$E_{tot} = \int_0^1 \left( E_{ext}(v(s)) + \frac{1}{2} \left( \alpha(s) |v'(s)|^2 + \beta(s) |v''(s)|^2 \right) \right) ds \quad (2.23)$$

Un point quelconque de la courbe est représenté par :

$$v(s_i) = v_i = (x_i, y_i)^t \quad (2.24)$$

La dérivée première des coordonnées par rapport au paramètre  $s$  peut être approximée par différences finies et le carré de la norme du vecteur des dérivés premières devient :

$$\|v_i'\|^2 = \left\| \frac{dv_i}{ds} \right\|^2 \quad (2.25) \text{ par}$$

différences finies on trouve l'énergie de continuité :

$$\left\| \frac{dv_i}{ds} \right\|^2 = \|v_i - v_{i-1}\|^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \quad (2.26)$$

La dérivée seconde s'approxime par :

$$\|v_i''(s)\| = \left\| \frac{d^2 v_i}{ds^2} \right\|^2 \quad (2.27)$$

L'énergie de courbure dépend alors de :

$$\left\| \frac{d^2 v_i}{ds^2} \right\|^2 = \|v_{i-1} - 2v_i + v_{i+1}\|^2 = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \quad (2.28)$$

Il est à remarquer que ces deux expressions supposent deux hypothèses. La première est que les points soient placés, le long de la courbe, à une distance unité les uns des autres.

L'expression à minimiser, sans ajout de contraintes particulières, est :

$$E_{tot} = (\alpha(s)E_{continuité} + \beta(s)E_{courbure} + \gamma(s)E_{image}) ds \quad (2.29)$$

L'algorithme est itératif. On examine, à chaque itération, le voisinage de chaque point. On choisit le point du voisinage qui donne la plus faible valeur de l'énergie totale.

Après discrétisation, la minimisation de la distance entre les points implique une rétraction naturelle du contour. Williams et Shah [WIL, 92] proposent une variante qui évite cette rétraction trop importante. Ils utilisent la différence de distance entre les deux points  $\|v_i - v_{i-1}\|$ , par rapport à la distance moyenne des points du contour  $\bar{d}$ . [ROU, 03]

$$E_{continuité} = \bar{d} - \|v_i - v_{i-1}\| \quad (2.30)$$

Les différents membres de l'expression sont normalisés avant minimisation. Les différentes énergies sont normalisées par division par la valeur du maximum dans le voisinage, donnant ainsi une valeur appartenant à l'intervalle [-1,1].

## 5. Algorithmes génétiques

### 5.1 Introduction

La résolution du problème de minimisation de la fonctionnelle d'énergie ,ainsi que l'initialisation des contours actifs par les méthodes classiques, ne sont pas aisés.

Les algorithmes génétiques sont de plus en plus utilisés. Ils sont basés sur un phénomène naturel qui a fait ses preuves : *l'évolution*. Plus précisément, ils s'inspirent de l'évolution d'une population d'individus dans un milieu donné.

C'est en 1975 que J. Holland et son équipe adaptaient, pour la première fois, les algorithmes génétiques pour la résolution de problèmes d'optimisation, en faisant un parallèle entre un individu dans une population et une solution d'un problème parmi un ensemble de solutions.

Ces algorithmes permettent de calculer une solution acceptable à un problème dont l'espace de solutions est trop grand pour être parcouru dans un temps raisonnable.

Les algorithmes génétiques sont basés sur trois éléments principaux : la *sélection*, le *croisement* et la *mutation*. Dans la littérature on parle alors d'*opérateurs* de reproduction.

Leur principe est simple. Il est basé sur trois phases :

1. la genèse = l'initialisation aléatoire d'individus pour la population de la première génération.
2. la reproduction = l'évolution des individus de la génération courante vers la suivante.
  - (a) la sélection des individus reproducteurs.
  - (b) le croisement génétique de ces individus pour la création de nouveaux individus.
  - (c) la mutation de certains individus pour que le pool génétique ne s'affaiblisse pas.
  - (d) l'évaluation des individus par le calcul de leur fitness.
3. recueil du meilleur individu = recherche de l'individu le plus adapté selon les critères souhaités.

Les sous étapes 2 et 3 se répètent autant de fois qu'il y a besoin de générations (itérations de l'algorithme) pour satisfaire un critère d'arrêt. Celui-ci est défini avant que le processus ne commence. La solution est alors représenté par le meilleur individu de la dernière génération.

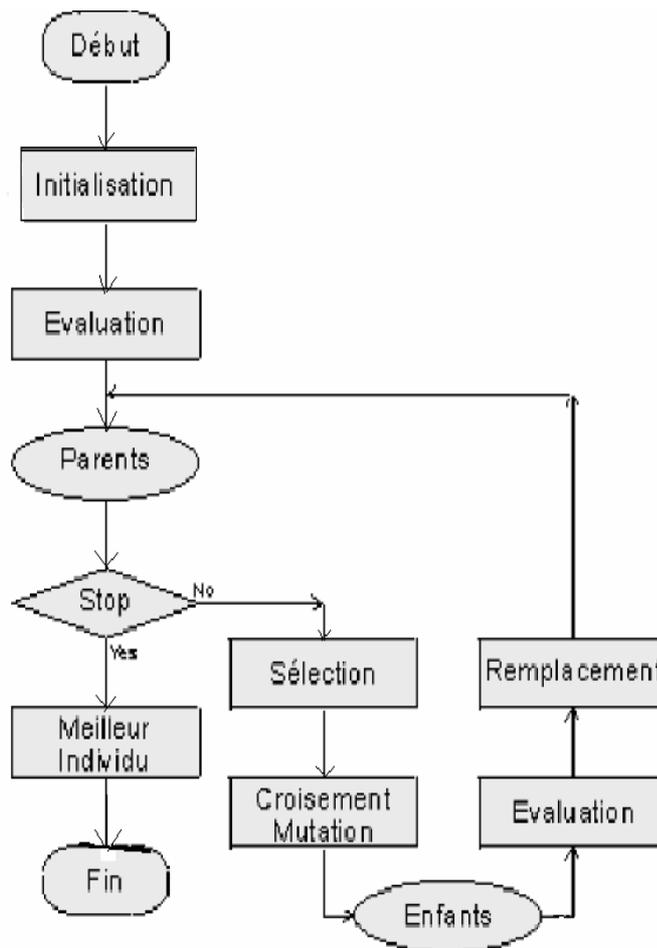


Figure 2.3 - Principe de base d'un algorithme génétique.

La mise en oeuvre des algorithmes génétiques nécessite plusieurs étapes à détailler. La première est le *codage* de l'individu représenté par un chromosome. La seconde est de trouver la *fitness* (fonction d'adaptation). La troisième est de définir les *opérateurs* de la reproduction (c-à-d la sélection, le croisement et la mutation).

## 5.2 Le codage

Le *codage* est une partie très importante des algorithmes génétiques. Il permet de représenter l'individu sous la forme d'un chromosome. Ce chromosome est constitué de gènes qui prennent des valeurs dans un alphabet binaire ou non.

Le choix du codage est délicat. Il doit permettre de coder toutes les solutions et permettre la mise en oeuvre des opérateurs de reproduction. C'est ainsi que le bon déroulement des algorithmes génétiques sera assuré.

Plusieurs codages sont employés. Voici quelques exemples.

### 5.2.1 Le codage binaire

Le gène est codé par un caractère binaire, 0 ou 1. C'est le plus courant et celui qui a été employé

lors de la première application des algorithmes génétiques.

**Exemples:**

Chromosome A 

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

Chromosome B 

1	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

**5.2.2 Le codage par permutations de valeurs entières**

Le gène est codé par une valeur entière dans un ensemble de cardinalité égale au nombre de gènes. Ce codage est bien adapté aux problèmes d'ordonnement.

**Exemples :**

Chromosome A 

1	3	6	2	5	8	4	7
---	---	---	---	---	---	---	---

Chromosome B 

6	1	3	2	7	4	5	8
---	---	---	---	---	---	---	---

**4.2.3 Le codage par valeurs**

Le gène est codé par une valeur prise dans un ensemble fini ou infini. Ce codage est généralement utilisé pour des valeurs qu'on ne peut pas mettre sous la forme d'un des deux codages précédents. Ces valeurs sont bien entendu liées au problème à résoudre.

**Exemples**

Chromosome A 

A	A	C	G	T	T	C	G
---	---	---	---	---	---	---	---

Chromosome B 

2,34	5,67	0,50	0,34	1,87	5,78	3,67	9,12
------	------	------	------	------	------	------	------

Chromosome C 

BLUE	RED	GREEN	BLUE	WHITE	RED	RED	BLACK
------	-----	-------	------	-------	-----	-----	-------

**5.3 La fonction d'adaptation**

La fonction d'adaptation, ou *fitness*, associe une valeur pour chaque individu. Cette valeur a pour but d'évaluer si un individu est mieux adapté qu'un autre à son environnement. Ce qui signifie qu'elle quantifie la réponse fournie au problème pour une solution potentielle donnée.

Ainsi les individus peuvent être comparés entre eux.

Cette fonction, propre au problème, est souvent simple à formuler lorsqu'il y a peu de

paramètres. Au contraire, lorsqu'il y a beaucoup de paramètres ou lorsqu'ils sont corrélés, elle est plus difficile à définir. Dans ce cas, la fonction devient une somme pondérée de plusieurs fonctions. Un ajustement des coefficients est alors nécessaire.

## 5.4 Les opérateurs de reproduction

Les algorithmes génétiques sont basés sur un phénomène naturel : l'évolution. Plus précisément, ils supposent, qu'a priori, deux individus adaptés à leur milieu donnent, par recombinaison de leurs gènes, des individus mieux adaptés. Pour ce faire, trois opérateurs sont à disposition : la sélection, le croisement et la mutation, plus un opérateur optionnel, l'élitisme.

### 5.4.1 La sélection

La sélection sert à choisir dans l'ensemble de la population les individus qui participeront à la reproduction. Ainsi les meilleurs individus ont plus de chance de survivre et de se reproduire. Nous présenterons les deux plus connus.

#### 5.4.1.1. La roulette

Cette méthode exploite la métaphore d'une roulette de casino. La roue est divisée en autant de secteurs que d'individus dans la population. La taille de ces secteurs est proportionnelle à l'adaptation de chaque individu. En faisant tourner la roue, l'individu pointé à l'arrêt de la boule est sélectionné. Les individus les mieux adaptés ont donc plus de chance d'être tirés au sort lors du déroulement du jeu.

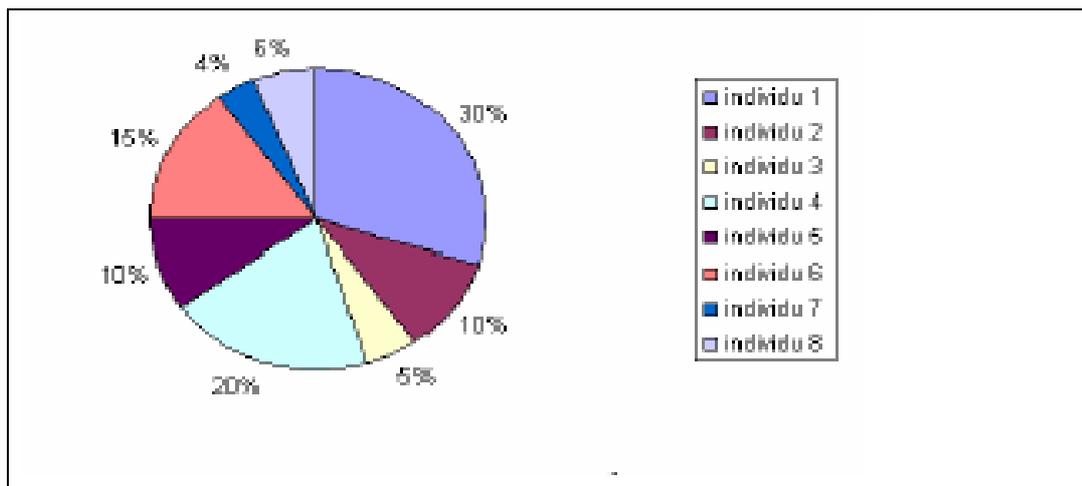


Figure 2.4 : La roulette

#### 5.4.1.2 Le tournoi

Cette méthode ressemble plus à ce qui se passe dans la réalité. Comme son nom l'indique, elle fait s'affronter deux ou plusieurs individus afin que le meilleur gagne. Plusieurs variantes existent. On peut par exemple faire varier le nombre d'individus qui doivent s'affronter au départ, ou encore permettre ou non que le même individu soit éligible plusieurs fois lors d'un même tournoi.

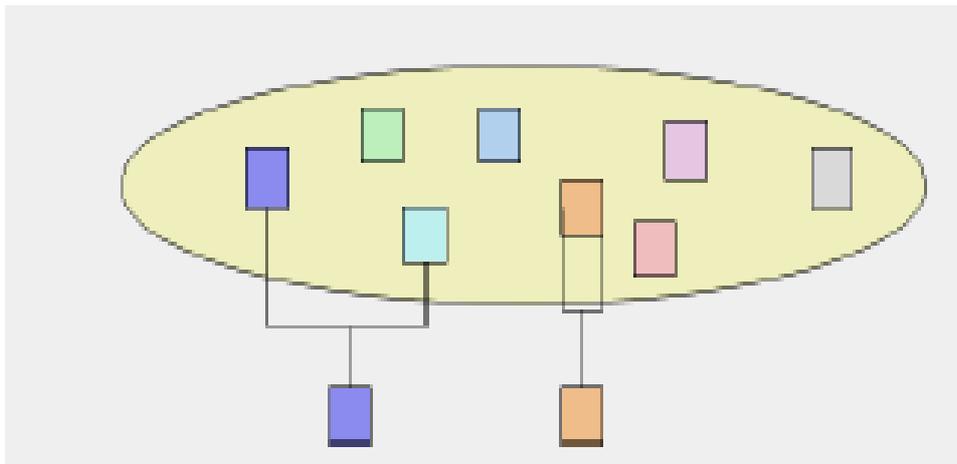


Figure2.5 : Le tournoi entre deux individus avec rééligibilité.

## 5.4.2 Le croisement

Le croisement combine les gènes des deux individus parents pour donner deux nouveaux chromosomes d'individus enfants. La zone de croisement est généralement choisie aléatoirement dans les chromosomes. Les méthodes de croisement sont liées au codage. Le croisement de chromosomes codé en binaire ne sera pas le même que celui d'un chromosome codé par valeur entière mais leur principe est identique.

### 5.4.2.1 Le croisement à un point de coupure.

Dans le chromosome, un point de croisement est choisi. La première partie du chromosome de l'individu dit parent 1 est alors copiée sur un individu de la prochaine génération, enfant 1, celle du parent 2 est copiée sur un enfant 2. Pour la deuxième partie du chromosome, les parents échangent leurs enfants, ainsi le parent 1 est copié sur l'enfant 2 et le parent 2 sur l'enfant 1.

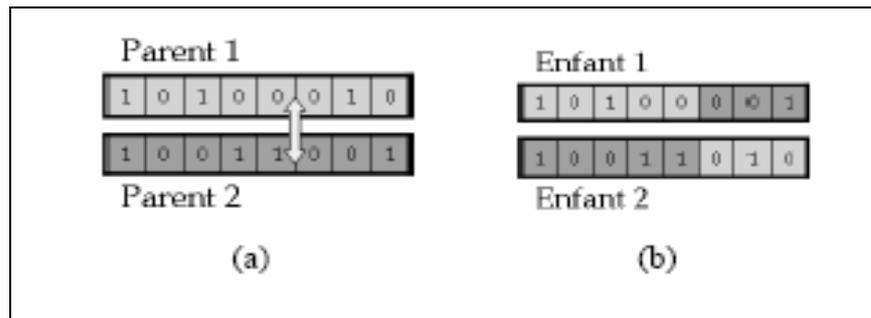


Figure 2.6 : Le croisement d'un point.

#### 5.4.2.2 Croisement uniforme

Dans le cas du croisement uniforme, on utilise un masque binaire aléatoire de même longueur que le génome d'un chromosome. Si le bit  $i$  du masque est à 1, alors l'enfant 1 prend le bit  $i$  du parent 1, et l'enfant 2 prend le bit  $i$  du parent 2. Si par contre c'est un 0, c'est l'inverse qui est effectué. Le principe est décrit comme suit.

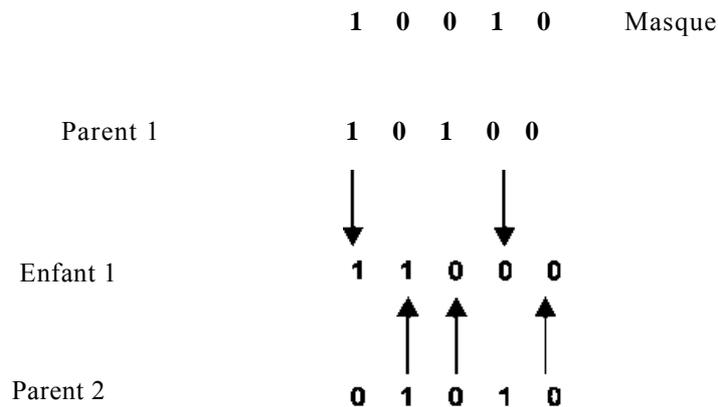


Figure 2.7 : Croisement uniforme

#### 5.4.3 La Mutation

Comme les individus les mieux adaptés sont les plus susceptibles d'être choisis lors de la sélection, la perte de certains gènes est inévitable avec le temps.

La mutation est l'opérateur qui permet d'éviter la dégénérescence de la population et d'enrichir le pool de gènes. Cette dégénérescence peut se traduire par une convergence des individus vers un optimum local, d'où l'importance de la mutation.

Classiquement, la mutation modifie aléatoirement, un petit nombre de gènes, avec un faible taux de probabilité.

### 5.4.3.1 La mutation avec le codage binaire

Un ou plusieurs gènes, selon la taille du chromosome, et un taux  $\frac{\text{gènes mutés}}{\text{gènes total}}$  très faible sont choisis aléatoirement. Il sont alors inversés (1  $\rightarrow$  0 et 0  $\rightarrow$  1).

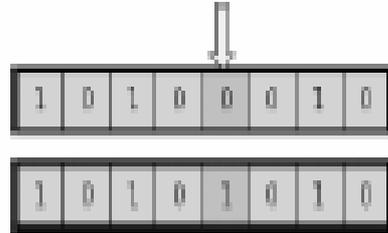


Figure 2.8 : La mutation avec le codage binaire.

### 5.4.3.2 La mutation avec le codage par valeurs entières

Le principe est le même que précédemment, plusieurs gènes sont choisis et permutés.

Pour le codage par valeur entière, le taux gènes  $\frac{\text{gènes mutés}}{\text{gènes total}}$  étant plus important (car un nombre plus grand de gènes sont mutés), il faut un taux de mutation plus faible que dans le cas d'un codage binaire.

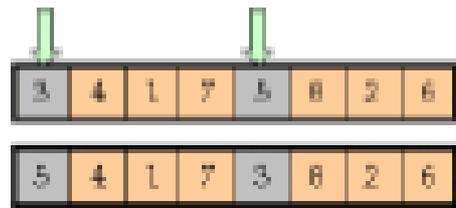


Figure 2.9 : La mutation avec le codage par valeurs entières

### 5.4.4. L'élitisme

L'élitisme est un opérateur optionnel particulier. Il permet de garder l'individu le mieux adapté d'une génération à la suivante. En effet, l'opérateur de sélection peut ne pas le sélectionner, le croisement avec un autre individu peut donner des individus moins adaptés si les gènes ne sont pas bien re-combinés ou encore sa mutation peut également le rendre moins adapté.

**Conclusion**

Cette partie nous a permis de mieux comprendre l'intérêt de la méthode des contours actifs dans la recherche d'une courbe de segmentation dans une image et d'étudier la puissance exploratoire des algorithmes génétiques. Nous pouvons conclure que les AG sont des algorithmes simples de conception et peuvent résoudre des problèmes assez complexes. La résolution de ces problèmes est obtenue grâce aux opérateurs de reproduction (sélection, croisement et mutation).

## Chapitre 3

### Suivi d'objets en mouvement

#### 1. Introduction

L'étude de séquences d'images numériques est un domaine qui suscite une attention toute particulière en analyse d'images. En effet les investigations portent sur : *l'extraction d'informations sur le mouvement*. Ces informations peuvent concerner les domaines comme la *détection* de mouvement, *l'estimation* du mouvement, la *segmentation* au sens du mouvement ou encore le suivi des objets en mouvement [BOS,00].

#### 2. Mouvement (déplacement)

Le suivi d'un objet en mouvement dans une séquence d'images, également appelé poursuite de cible, consiste à estimer l'état de cet objet (ou cible) sur chaque image de la séquence à partir d'observations. L'état de l'objet regroupe un ensemble de caractéristiques descriptives comme par exemple sa position, sa vitesse ou encore sa forme. Son estimation s'effectue à partir d'observations ou mesures issues de l'image [FOR,03].

On distingue deux types de Mouvement.

##### 2.1 Mouvement d'un objet rigide

Communément, un objet rigide est un objet solide qui garde toujours sa forme. Par abus de langage, lorsqu'un tel objet se déplace, tous les points qui le composent se déplacent de façon identique.

##### 2.2. Mouvement d'un objet déformable (non rigide)

Un objet déformable est un objet qui n'est pas rigide. Donc, lorsqu'il est en mouvement, chacun de ses points peut subir une transformation différente.

#### Exemple : boule de pétanque /boule anti-stress

A titre d'exemple de comparaison, ces deux types de mouvement peuvent être symbolisés par l'opposition entre une boule de pétanque et une boule anti-stress.

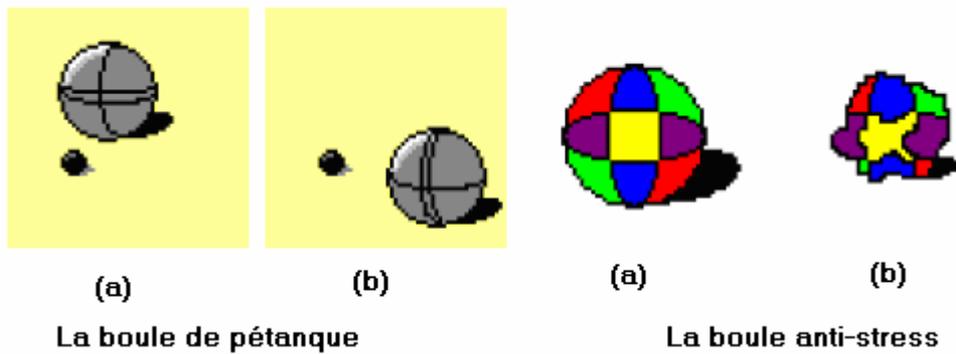


Figure 3.1 : Exemple : boule de pétanque et boule anti-stress

### 3. Estimation du Mouvement

#### 3.1. Mouvement apparent

Le mouvement apparent est le mouvement qui est observé sur l'image. Il est représenté par le champ mouvement apparent ou le flux optique.

Bien que ce mouvement soit toujours observé, il n'est pas toujours dû à une réalité dans le repère monde. En effet, l'estimation d'un mouvement qui n'existe pas réellement peut avoir lieu.

L'estimation du mouvement apparent porte uniquement sur ce qui est observé [ALE, 97].

#### 3.2 Mouvement réel 2-D

Le mouvement réel est le mouvement qui se produit dans l'espace monde. Il est représenté par les champs de déplacement 2-D ou les champs de vitesses.

L'inconvénient majeur de ce mouvement est qu'il n'est pas toujours observable lors du passage à l'image. En effet, ce qui est observable dans une image est une différence de luminosité, or tous les mouvements ne génèrent pas cette différence d'une image à l'autre.

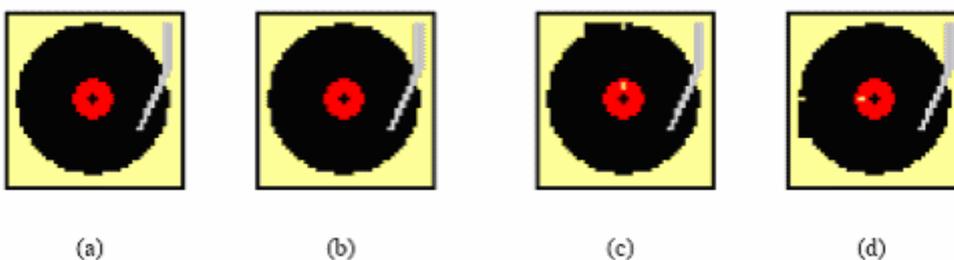


Figure 3.2 : Exemple de séquences de l'image disque

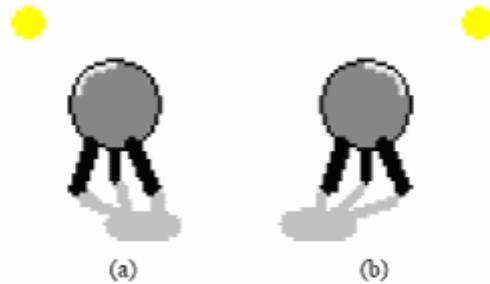


Figure 3.3 : séquences de l'image statue

#### 4. Segmentation par le mouvement

L'apport de la vidéo dans l'analyse de scène est certainement la possibilité d'utiliser le mouvement comme information supplémentaire à celle de la couleur [BED, 94].

Les techniques de détection de mouvement peuvent être classées de différentes façons :

- celles basées sur des différences d'images.
- celles utilisant un test de vraisemblance.
- celles qui améliorent les décisions ponctuelles par des traitements spatiaux.
- celles qui prennent explicitement en compte les variations d'éclairement.

#### 5. Suivi par contour actif (snake)

Dans cette partie, on va utiliser l'approche de segmentation par contour actif qui permet d'utiliser un modèle de contour actif, mais ne procède à aucun filtrage ou segmentation précédant la déformation du contour, ce qui permet de diminuer le temps de calcul [LEF,01].

Le modèle du snake tel qu'il est décrit précédemment peut être utilisé aussi bien pour localiser précisément un contour dans une image fixe que pour suivre un mouvement non rigide du contour.

Le but est de suivre des objets rigides ou non rigides dans une scène en mouvement. Afin de réduire le temps de calcul, aucun prétraitement n'est nécessaire à la déformation du contour actif et aucune estimation de mouvement n'est effectuée pour le suivi. La méthode se décompose en deux parties, exécutées successivement pour chaque image :

- l'initialisation sous forme de rectangle
- puis la déformation du contour actif.

### 5.1 Initialisation sous forme de rectangle

Les algorithmes de suivi d'objet permettent de trouver la nouvelle position à partir d'une position courante. Il est donc nécessaire à l'initialisation de leur fournir la première position de la cible. Le plus simple est de laisser l'utilisateur placer celle-ci à un endroit prédéfini connu de l'algorithme de suivi qui est ensuite déclenché manuellement [LEJ, 01].

Pour une image donnée, l'initialisation par un rectangle est effectuée en deux étapes. Tout d'abord un rectangle parallèle aux contours de l'image est créé. Sa position et sa taille sont déterminées de telle sorte qu'il englobe le contour actif final obtenu à l'image précédente.

La taille de ce rectangle est ensuite augmentée afin d'englober le contour actif final supposé pour l'image courante.

Le contour actif initial est obtenu en plaçant les points qui le composent de manière uniforme sur le contour du rectangle.

Un traitement spécifique est nécessaire pour la première image d'une séquence, ou a priori aucune information sur le contour n'est disponible. Actuellement, l'initialisation sous forme de rectangle est remplacée par une initialisation manuelle du contour effectuée par l'utilisateur ou une initialisation automatique obtenue par une détection des objets en mouvement.

### 5.2 Déformation du contour actif

Le contour actif, dont le nombre de points dépend de la précision voulue, est ensuite déformé et se fixe autour de l'objet. Il est réduit jusqu'à convergence avec les forces décrites précédemment.

Pour trouver la position de la cible du suivi dans une nouvelle image, le snake est initialisé sur sa position dans l'image précédente et le processus de minimisation est relancé. L'idée d'utiliser le snake comme technique de suivi était déjà présente dans [KAS, 87].

Cependant, si entre les deux images le déplacement de la cible est trop grand, le snake la perd et se stabilise sur un minimum local de l'image la plus proche. C'est pourquoi le modèle de base des Snakes a été adapté à l'objectif particulier du suivi d'objet.

**Conclusion**

Le problème du suivi d'objet dans une séquence d'images est résolu par la méthode des contours actifs. Cette méthode consiste en l'initialisation par un rectangle, puis la déformation en utilisant les différentes énergies intervenant dans le processus de convergence.

Nous avons implémenté un algorithme de suivi d'objets par les méthodes de Greedy et Greedy autonome dont l'organigramme et les résultats seront présentés dans le prochain chapitre.

# Chapitre 4

## Implémentation des algorithmes

### 1. Introduction.

Nous avons présenté à travers les chapitres précédents, les différentes méthodes de segmentation, mais plus particulièrement les contours actifs. Dans ce présent chapitre, étant donné que notre travail se concentre sur les méthodes de Greedy, de Greedy autonome et les algorithmes génétiques, nous verrons la manière avec laquelle ces derniers ont été implémentés. Rappelons que les contours actifs se composent de différentes étapes qui sont l'initialisation, la discrétisation et la détection des contours. Alors nous verrons pour chaque étape, quelles sont les méthodes adéquates, choisies parmi celles énoncées dans les chapitres précédents, à implémenter.

### 2. Transformation de l'image en niveaux de gris.

Vu que dans notre cas les filtres et algorithmes sont implémentés pour des images à 256 niveaux de gris. Nous convertissons les images couleur en niveaux de gris. Les images en couleur, possédant trois composantes de couleur : Rouge Vert et Bleu, sont transformées en images possédant une seule composante grise, représentative des trois anciennes composantes pour ne pas détériorer trop fortement l'image et donc les contours. La valeur de cette dernière est calculée selon la relation :

$$\text{Gris} = (0,299*\text{Rouge}+0,587*\text{Vert}+0,114*\text{Bleu}) \quad (4.1)$$

### 3. Prétraitement, lissage de l'image.

Avant tout traitement de détection de contours, il est nécessaire de lisser l'image. Cela conduit à réduire le bruit qui dégrade considérablement la qualité de l'image.

On a choisi parmi les différents types de filtres présentés dans la partie état de l'art, d'implémenter des filtres linéaires (le filtre moyen). En pratique l'opération de filtrage consiste à multiplier l'image par un masque qui est une matrice à trois dimensions ou bien à cinq dimensions.

**Exemple de filtre moyen :**

$$\frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ Filtre moyen } 3 \times 3 ; \quad \frac{1}{25} \times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ Filtre moyen } 5 \times 5$$

Figure 4.1: Masque du filtre Moyen

**Pour tout point  $I_{i,j}$  de l'image I**

$$I_{i,j} = (I_{i-1,j-1} + I_{i-1,j} + I_{i-1,j+1} + I_{i,j-1} + I_{i,j} + I_{i,j+1} + I_{i+1,j-1} + I_{i+1,j} + I_{i+1,j+1}) / 9$$

Figure 4.2: Exemple de lissage par un filtre moyen 3x3

### 3.1 Opérateurs gradients

Pour obtenir le gradient d'une image, chaque pixel, chaque case de la matrice image sera calculé en fonction de ses voisins pondérés par une matrice de coefficients déterminés. On peut utiliser les méthodes de Roberts, Prewitt ou de Sobel.

#### 3.1.1 Méthode de SOBEL

Pour appliquer un opérateur de SOBEL, on commence à calculer les deux gradients selon x et selon y en chaque point de l'image. Cela se fait par un produit de convolution entre l'image et chacune des deux matrices de la figure 4.3.

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Masque en X Masque en Y

Figure 4.3: Masque de SOBEL

Les gradients de SOBEL d'une image selon X et selon Y notés respectivement  $\nabla \hat{I}_x$  et  $\nabla \hat{I}_y$ , sont les résultats du produit de convolution effectué précédemment, ils représentent les dérivées premières de l'image  $I$  lissée, notée  $\hat{I}$ .

Pour obtenir le gradient  $\nabla I$ , on applique le Gradient de SOBEL en suivant la procédure de calcul du gradient de SOBEL

$$\nabla I = \sqrt{\nabla \hat{I}_x^2 + \nabla \hat{I}_y^2} \quad (4.2)$$

#### 4. Contours actifs et algorithme de Greedy.

On peut illustrer notre implémentation par l'organigramme ci-dessous figure 4.4, montrant un schéma de conception de la segmentation d'images par les contours actifs avec minimisation et déformation suivant l'algorithme de Greedy.

##### 4.1. Initialisation du contour.

L'initialisation du contour se fait par le dessin d'un rectangle, directement sur l'image, qui sera notre contour initial à  $t=0$ . Les coordonnées des pixels composant ce rectangle seront récupérés dans la structure de donnée représentant le contour. La structure utilisée est une liste contenant en chaque nœud les coordonnées d'un point du contour. On récupère les points du carré. Selon une distance entre les points choisis par l'utilisateur. La sélection des pixels à prendre se fait d'une manière uniforme.

Il est important que les pixels soient récupérés selon leurs adjacences, pour les besoins de rigidité de la courbe.

Le contour initial (le rectangle) en pratique est représenté par deux points. Le premier point est celui du coin supérieur gauche, et le deuxième point est celui du coin inférieur droit. Comme représenté par la figure 4.5.

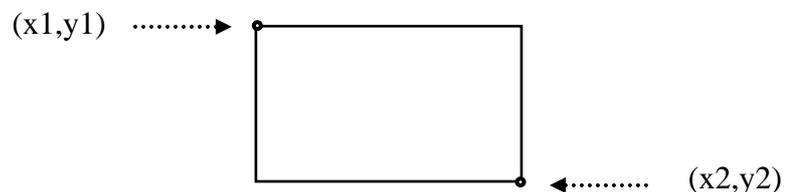


Figure 4.5: les deux points représentant la courbe initiale

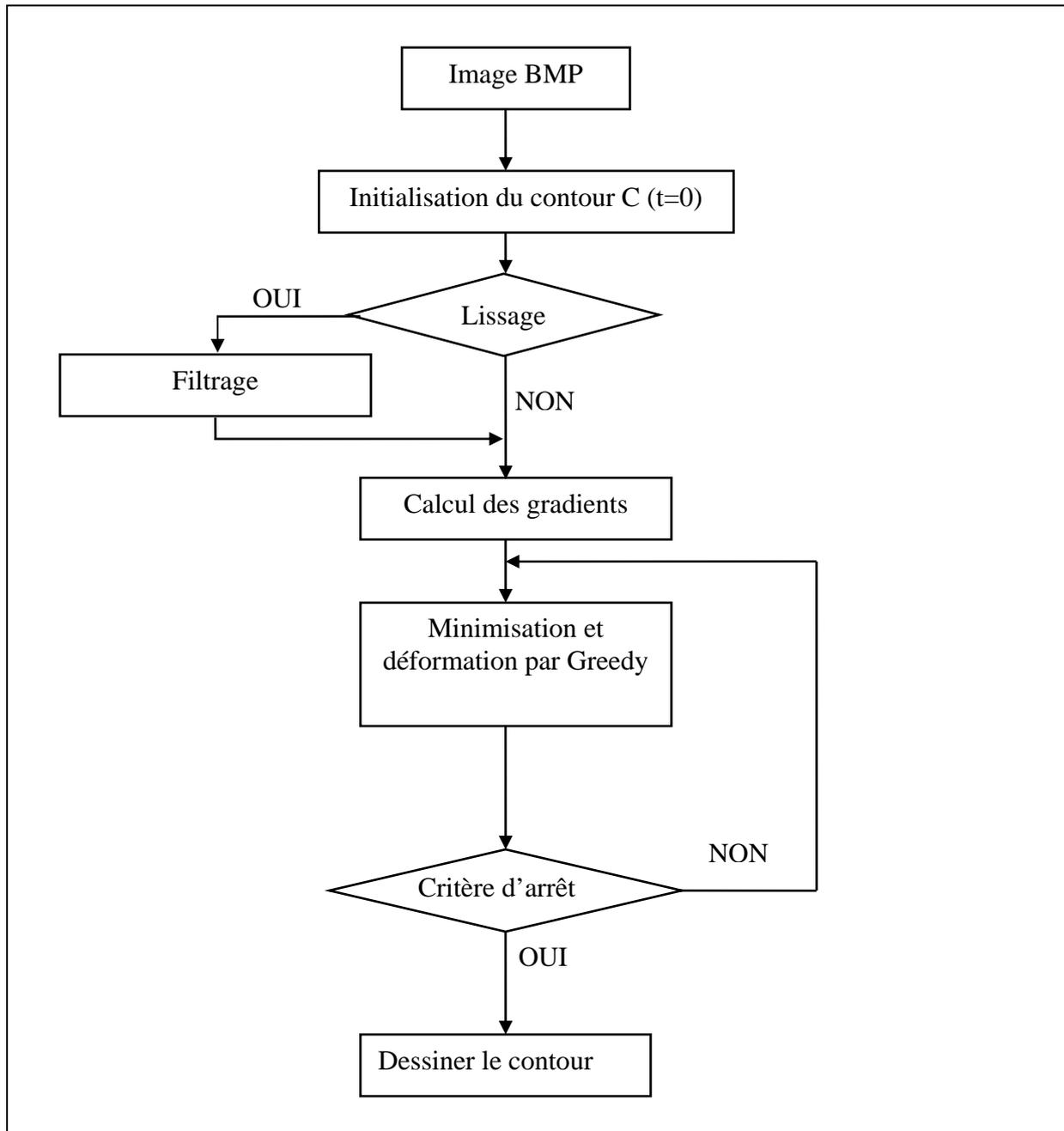


Figure 4.4 : Organigramme de l'algorithme de Greedy .

## 4.2. Prétraitements

Avant d'entamer l'étape de minimisation et déformation du contour, l'image peut être débarrassée du bruit, en appliquant un des filtres de lissage déjà présenté. Après le lissage, et dans le but d'éviter le calcul des gradients à chaque étape de l'algorithme, on a préféré de ne les calculer qu'une seule fois. Les gradients sont sauvegardés dans une matrice et sont récupérés à la demande. Les gradients sont calculés par l'opérateur de Sobel.

## 4.3. Minimisation et déformation par l'algorithme de Greedy :

### 4.3.1. Calcul des énergies.

Pour chaque point du *contour actif*  $C$ , on choisit un nombre de voisins pour les quels on va calculer l'énergie ; le point se déplacera alors sur le voisin qui possède l'énergie la plus faible. On cherche donc l'ensemble  $M$  des points pour lequel l'énergie totale est minimale.

$$E(C) = \sum_{i=1}^N \left( aE_{\text{continuité}}(M_i) + bE_{\text{courbure}}(M_i) + cE_{\text{intensité}}(M_i) + dE_{\text{gradient}}(M_i) + eE_{\text{ballon}}(M_i) \right) \quad (4.3)$$

avec  $M_i$   $i = 1..N$ , un ensemble de points ordonnés

#### 4.3.1.1. L'énergie de Continuité.

L'énergie de continuité fait partie des énergies dites internes au contour actif. Cette énergie régit la distance entre les différents points du contour actif. Ainsi, lorsque le coefficient de l'énergie de continuité est nul, les points du contour actif pourront se déplacer aussi loin les uns des autres que l'image leur permettra. Dans le cas contraire, c'est-à-dire lorsque le coefficient de continuité est très élevé, le contour actif sera rigide.

Du fait de la forte valeur du coefficient de continuité, la distance entre les points du contour actif est fixe. Cela oblige les points du contour actif à se déplacer à une distance fixe de ses points adjacents.

#### **4.3.1.2. L'énergie de Courbure.**

L'énergie de courbure définit l'angle formé par trois points adjacents du contour actif. Plus le coefficient attribué à cette énergie sera fort, et plus le contour actif aura la forme connue pour être la forme la moins « coûteuse » en énergie : la ligne droite ou le cercle pour des objets fermés.

#### **4.3.1.3. L'énergie d'Intensité.**

L'énergie d'Intensité fait partie des énergies externes. Cette énergie représente simplement la valeur du pixel (sur un octet, donc de 0 à 255) de l'image transformée en niveaux de gris.

#### **4.3.1.4. L'énergie de Gradient.**

L'énergie de Gradient est la deuxième énergie du contour actif qui dépende de l'image.

Cette énergie externe est d'une importance première pour la détection du contour. En effet, un contour est généralement caractérisé par une forte différence entre les valeurs de plusieurs pixels. Ainsi, si la dérivée d'une fonction représente les pentes d'une courbe, le gradient montrera les fortes différences, les contours de l'image.

#### **4.3.1.5. L'énergie de Ballon.**

L'énergie de Ballon est l'énergie qui décide du sens de propagation du contour actif. Un coefficient d'énergie de ballon positif va concentrer le contour actif, alors qu'un coefficient négatif va rendre le contour actif expansif.

### **4.3.2. Parcours des différents paramètres supplémentaires réglables**

#### **4.3.2.1. Le voisinage.**

Pour déplacer un point, l'algorithme de Greedy calcule toutes les énergies des pixels présents dans le voisinage du pixel où le point est.

Pour un voisinage de trois par exemple, l'algorithme va calculer les énergies d'une matrice de 7x7 pixels pour choisir son déplacement

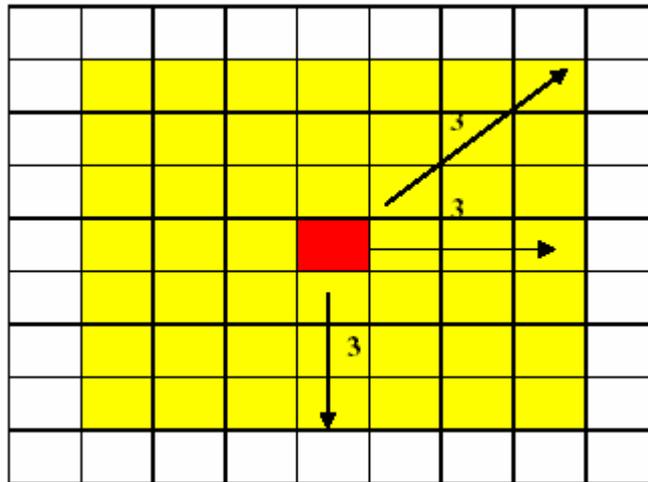


Figure 4.6 : Exemple d'un voisinage de 3

Ce paramètre du Greedy permet au contour actif de se déplacer assez rapidement, en terme d'itérations.

#### 4.3.2.2 La normalisation.

Afin d'éviter une trop grande dispersion des valeurs des énergies (certaines énergies varient entre 0 et 500 alors que d'autres entre -1 et 0), il est absolument nécessaire de normaliser, c'est-à-dire de diviser la valeur calculée par la plus grande valeur du voisinage.

Cela permet ainsi, d'obtenir des variations comprises entre -1 et 1, pour toutes les énergies.

#### 4.3.2.3. Les critères d'arrêt.

Les critères d'arrêt sont très importants. En effet, il faut préciser au contour actif son arrêt, dans certains cas, le contour actif s'arrête pendant un moment sur les contours de l'objet puis dépasse ceux-ci, attiré par des voisins dont l'énergie est plus faible .ou bien il fait des itérations supplémentaires inutiles.

Le contour actif s'arrête lorsque :

- Il atteint le nombre d'itérations introduit par l'utilisateur.
- La stabilité de l'énergie totale ( $E(C)$  à l'itération  $i$  égale  $E(C)$  à l'itération  $i + 1$ ).

#### 4.3.3 Mouvement du contour.

La méthode de *Greedy* consiste à faire évoluer le contour actif en minimisant la fonction d'énergie : pour chaque point du *contour actif*, on choisit un nombre de voisins pour les quels on va calculer l'énergie ; le point se déplacera alors sur le voisin qui possède l'énergie la plus faible. On cherche donc l'ensemble  $M$  des points pour laquelle l'énergie est minimale. Donc on déplace

un point unique pour constituer un nouveau contour actif à chaque itération. Tous les points sont traités successivement lors de chaque itération.

La figure 4.7 nous montre la procédure avec laquelle on fait évoluer les points du contour actif à chaque itération

```
Début  
Faire  
  / Pour tous les points du snake  
  / / Pour tous les points du voisinage  
  / / / Calculer les énergies  
  / / Fin Pour  
  / / Pour tous les points du voisinage  
  / / / Normalisation  
  / / Fin Pour  
  / / Minimiser pour obtenir le nouveau point  
  / Fin Pour  
Jusqu'au critère d'arrêt  
Fin
```

Figure 4.7 : Algorithme de la déformation par la méthode de Greedy.

#### 4.3.4 Extraction de la courbe.

Le dessin du contour sur l'image se fait de manière continue, les points adjacents appartenant à la courbe sont liés par une ligne directe. L'extraction de la courbe finale se fait par une procédure implémentée [HUA,03].

Procédure Dessiner\_contour

*Pour chaque point  $i$  du contour  $Point_i$*

Tracer\_ligne de Couleur Rouge ( $Point_i, Point_{i+1}$ )

*Fin*

Procédure d'extraction de la courbe

## 5. Contours actifs et algorithme de Greedy Autonome.

Comme on l'a vu précédemment, un contour actif est composé de plusieurs énergies, auxquelles on applique différents coefficients pour savoir où se déplacer. Le grand handicap de cette méthode est qu'il est nécessaire de régler, d'ajuster les coefficients pour obtenir de bons résultats, et cela peut prendre parfois beaucoup de temps. Aussi, un contour actif autonome a été conçu. Celui-ci génère en effet aléatoirement les coefficients d'énergie, et garde le meilleur jeu de coefficients pour les itérations suivantes.

Les étapes de prétraitement, d'initialisation du contour sont identiques à celles de Greedy simple.

### 5.1. Minimisation et déformation par l'algorithme de Greedy autonome :

Le calcul des énergies est identique à celui de Greedy simple sauf que les paramètres de pondération des énergies de chaque point de voisinage sont générés aléatoirement d'une façon autonome.

#### 5.1.1 Mouvement du contour.

Il s'agit de déterminer, pour chaque itération et pour chaque point du contour actif  $M$ , le meilleur jeu de coefficients qui minimisent la fonction d'énergie à cette itération.

Pour déterminer le meilleur jeu de coefficients, on tire aléatoirement un nouveau jeu de paramètres, à chaque point du contour actif, à chaque point du voisinage et à chaque itération.

Puis, en comparant la fonction d'énergie (calculée comme précédemment) avec ces coefficients et la précédente, on garde le nouveau jeu de coefficients si celui-ci livre une énergie plus faible.

La figure 4.8 montre la procédure avec laquelle nous faisons évoluer les points du contour actif à chaque itération

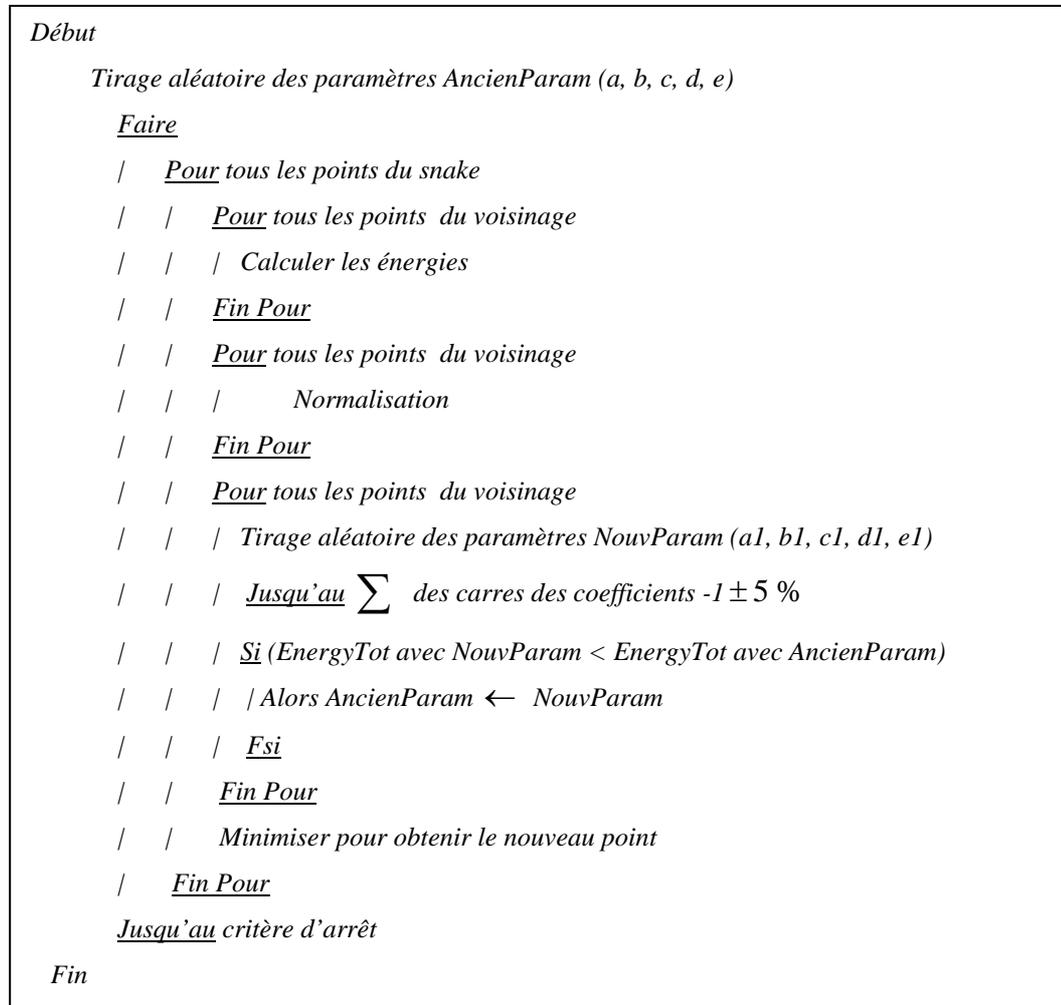


Figure 4.8 : Algorithme pour le contour actif autonome.

## 6. Suivi par contour actif (snake) :

Le modèle du snake tel que décrit précédemment peut être utilisé aussi bien pour localiser précisément un contour dans une image fixe que pour suivre le mouvement d'un objet rigide ou non rigide.

Afin de réduire le temps de calcul, aucun prétraitement n'est nécessaire à la déformation du contour actif et aucune estimation de mouvement n'est effectuée pour le suivi [VAL,01]. La méthode se décompose en deux parties, exécutées successivement pour chaque image :

- l'initialisation sous forme de rectangle
- puis la déformation du contour actif.

On peut illustrer notre implémentation par l'organigramme de la figure 4.9, montrant un schéma de conception de suivi d'un objet en mouvement par les contours actifs avec minimisation et déformation suivant l'algorithme de Greedy.

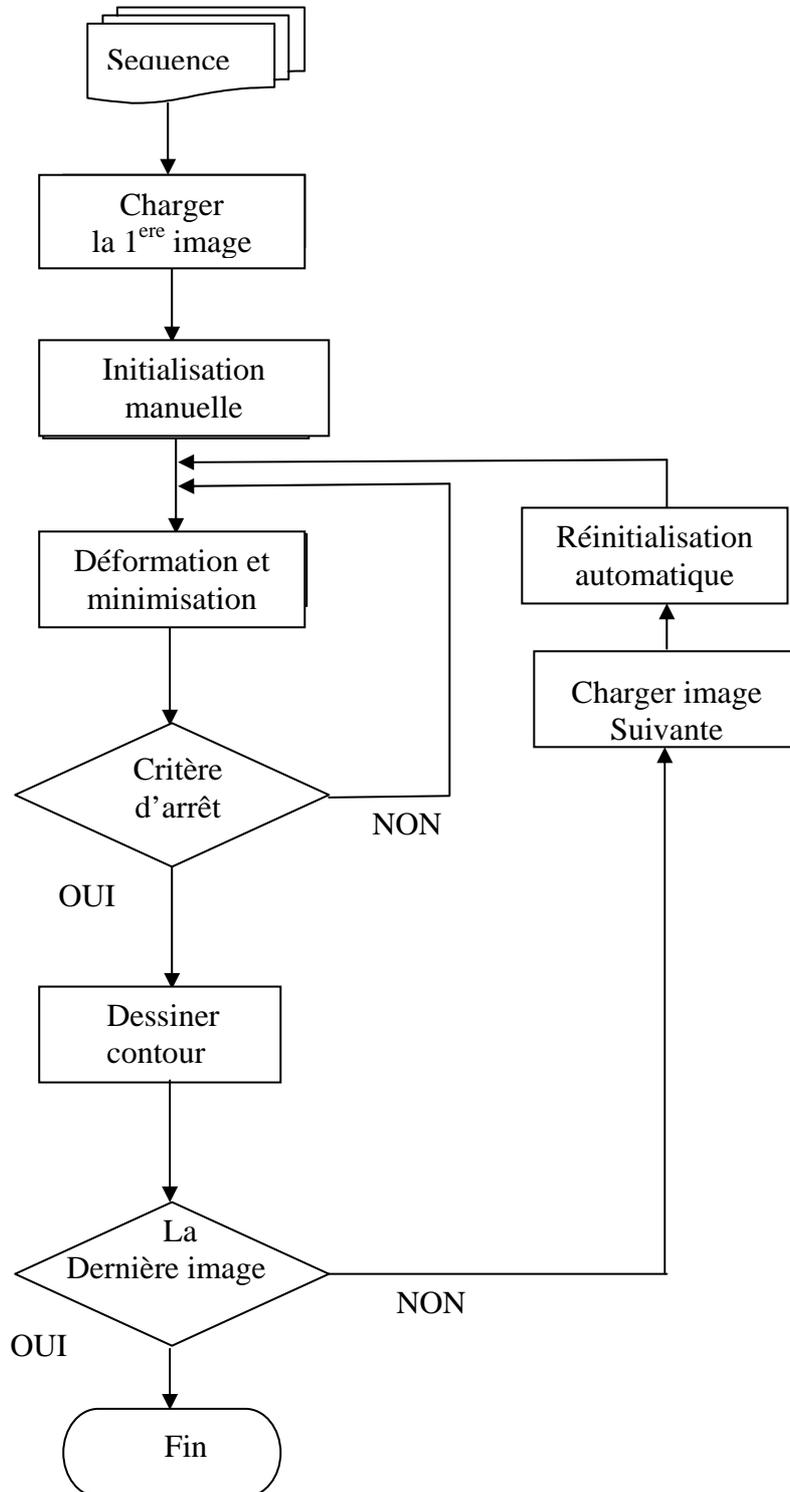


Figure 4.9 : Organigramme de suivi d'objet en mouvement par des contours actifs .

### 6.1 Initialisation du contour sur la première image.

Elle se fait comme en statique par le dessin d'un rectangle, directement sur l'image, qui sera le contour initial à  $t=0$ . Les coordonnées des pixels composant ce rectangle seront récupérées dans la structure de données représentant la courbe. La structure utilisée est une liste contenant en chaque nœud les coordonnées d'un point du contour. On récupère les points du carré, selon une distance entre les points choisis par l'utilisateur. La sélection des pixels à prendre se fait d'une manière uniforme.

### 6.2 Réinitialisation sous forme de rectangle.

Pour une image donnée, la réinitialisation par un rectangle est effectuée en deux étapes. Tout d'abord un rectangle parallèle aux contours de l'image est créé. Sa position et sa taille sont déterminées de telle sorte qu'il englobe le contour actif final obtenu à l'image précédente.

La taille de ce rectangle est ensuite augmentée afin d'englober le contour actif final supposé pour l'image courante.

La procédure avec laquelle nous faisons la réinitialisation automatique

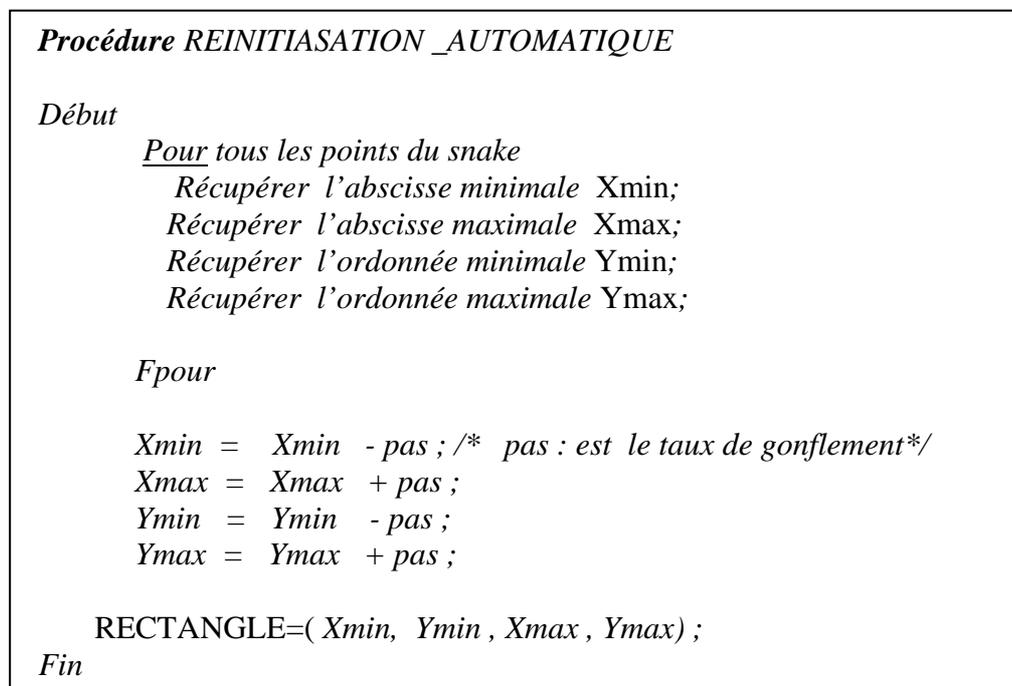


Figure 4.10 : Algorithme de réinitialisation automatique pour Greedy (resp.Greedy autonome) ;

Un traitement spécifique est nécessaire pour la première image d'une séquence, où a priori aucune information sur le contour n'est disponible. Actuellement, l'initialisation automatique sous forme de rectangle est remplacée par une initialisation manuelle du contour effectuée par l'utilisateur.

### 6.3 Déformation du contour actif.

Le contour actif, est ensuite déformé selon la méthode de Greedy (ou bien, la méthode de Greedy autonome) et se fixe autour de l'objet sélectionné. Il est réduit jusqu'à convergence avec les forces décrites précédemment (équation 4.3)

Pour trouver la position de la cible du suivi dans une nouvelle image, le snake est réinitialisé sur sa position dans l'image précédente et le processus de minimisation est relancé. Cependant, si entre les deux images le déplacement de la cible est trop grand, le snake la perd et se stabilise sur un minimum local de l'image la plus proche. Dans notre cas nous avons pris, après plusieurs essais, le taux de gonflement égal à deux intervalles ( $\text{pas} = 2$ ).

La figure 4.11 montre la procédure générale pour le traitement des séquences d'images

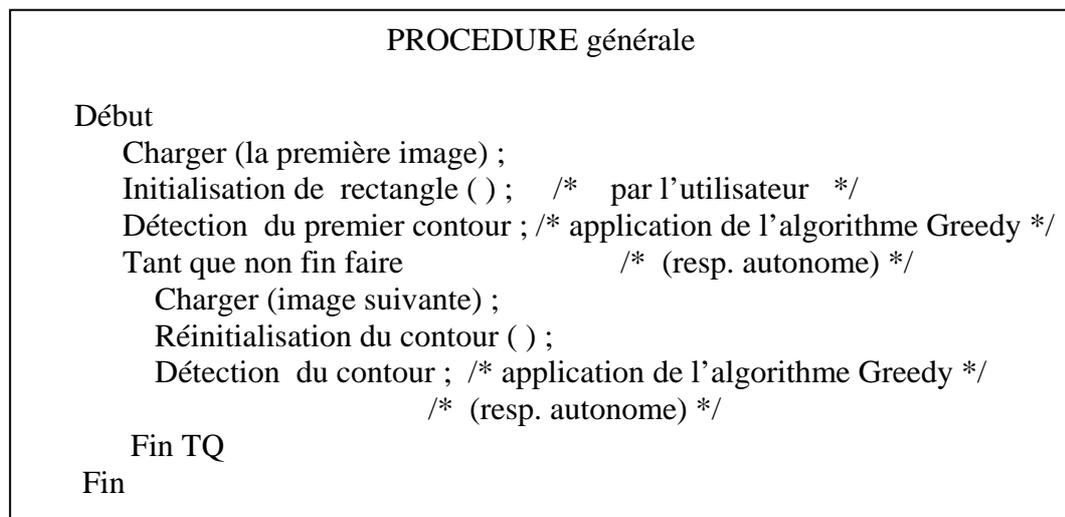


Figure 4.11: Procédure générale pour le traitement des séquences d'images

## 7. Implémentation par l'algorithme génétique.

On vient de voir l'implémentation de l'algorithme de Greedy. L'objectif de cette méthode est d'obtenir un optimum de la fonctionnelle suivante :

$$E(C) = \sum_{i=1}^N aE_{\text{intensité}}(M_i) + bE_{\text{continuité}}(M_i) + cE_{\text{courbure}}(M_i) + dE_{\text{gradient}}(M_i)$$

avec  $M_i, i = 1..N$ , un ensemble de points ordonnés. (4.4)

La difficulté est de définir une méthode qui, en un temps raisonnable, fournit une solution de qualité. La plupart des méthodes se trouvent piégées par des optimums locaux. Nous avons donc décidé de tester une méthode d'optimisation reconnue qui permet d'allier la puissance exploratoire des algorithmes génétiques et l'approche énergétique des contours actifs.

Le principe des algorithmes génétiques est de coder chaque solution potentielle d'un problème par un « chromosome ». L'ensemble des chromosomes ou « individus » forme alors la « population » qui va être amenée à évoluer au cours du temps. Une « génération » est l'état de la population à un instant  $t$ .

La population évolue au cours des générations en suivant des lois de sélection, de croisement et de mutation. En informatique, on parle d'opérateur génétique.

Dans l'algorithme génétique de base, il existe trois opérateurs dont les rôles sont représentés sur la Figure 2.3

L'opérateur de sélection donne plus de chance aux « bons » individus de participer à la génération suivante en fonction d'un certain critère, la *fitness*.

Le croisement mélange des chromosomes dans le but d'obtenir éventuellement de meilleurs individus.

La mutation effectue une modification des gènes des chromosomes des enfants.

En fonction du problème à résoudre, à chaque individu est associé un « degré d'adaptation à son environnement », C'est la *fitness*. Après croisements et mutations, une nouvelle génération est construite en conservant les individus de la population précédente ayant une propriété de *fitness* particulière ; cela permet la convergence vers une solution optimale.

La différence des algorithmes génétiques avec la plupart des algorithmes de contours actifs réside dans le fait que l'on ne fait pas évoluer un *snake*, mais une population de *snakes*. Seul le meilleur *snake* de la population à un instant donné est considéré comme l'approximation du contour. Si la taille de la population est suffisamment élevée, on peut se permettre d'initialiser les contours de manière totalement aléatoire dans l'image. C'est le jeu des croisements et des mutations qui permet de faire converger la population vers des individus intéressants. Cela permet de s'affranchir *a priori* de toute connaissance structurelle de l'image.

## 7.1 Les différentes étapes de l'algorithme.

Nous allons détailler dans ce paragraphe, les différentes étapes citées en figure 2.3.

### 7.1.1 Initialisation

La première étape de l'algorithme génétique est la construction d'une population initiale, un ensemble de chromosomes qui représentent des contours. Ces chromosomes doivent être déterminés aléatoirement, c'est-à-dire que pour chaque contour, est choisie aléatoirement une suite de points. Ces points sont ordonnés par l'ordre du tirage. Dans une image, généralement l'objet d'intérêt est relativement centré, pour réordonner les points, on a calculé leurs coordonnées polaires  $(\rho, \theta)$  ayant pour origine le centre  $(x_c, y_c)$  de l'image.

## Coordonnées polaires

Comme nous utilisons les coordonnées polaires pour trier les points, nous avons choisi de ne pas passer par les coordonnées cartésiennes, mais nous avons opté pour un codage directement en polaire des points dans le chromosome. Dans ce cas, l'angle  $\theta$  variera de 0 à  $2\pi$  et pour une image de largeur  $L$  et de hauteur  $H$  l'espace de définition devient :

$$\begin{aligned}
 \rho_{\max} &= \frac{L}{2 \cos \theta} & \text{si } \theta \in [0; \frac{1}{4}\pi] \cup [\frac{7}{4}\pi; 2\pi] \\
 \rho_{\max} &= \frac{H}{2 \sin \theta} & \text{si } \theta \in [\frac{1}{4}\pi; \frac{3}{4}\pi] \\
 \rho_{\max} &= \frac{-L}{2 \cos \theta} & \text{si } \theta \in [\frac{3}{4}\pi; \frac{5}{4}\pi] \\
 \rho_{\max} &= \frac{-H}{2 \sin \theta} & \text{si } \theta \in [\frac{5}{4}\pi; \frac{7}{4}\pi]
 \end{aligned} \tag{4.5}$$

Afin de réduire l'espace de recherche, sachant qu'on cherche un contour fermé plus ou moins centré sur l'image, on rend implicite la valeur de  $\theta$ . Les points sont répartis dans toutes les directions à partir de l'origine, c'est-à-dire pour un *snake* de 40 points, on prend  $v_i = (\rho_i, \theta_i)$ ;  $\theta_i = \frac{i}{20}\pi$ . Dans ce codage, seul  $\rho$  est explicitement codé ; cela réduit de moitié la longueur des chromosomes. L'espace de recherche n'est donc plus le plan de l'image mais l'union des 40 axes concourants.

### 7.1.2 Evaluation

C'est le calcul de la fitness ; on calcule l'énergie total pour chaque chromosome en utilisant la formule (4.4).

### 7.1.3 Parent

C'est l'ensemble des chromosomes (génération courante) à partir duquel nous allons faire la reproduction (avec les opérateurs génétiques).

## 7.1.4 Les opérateurs génétiques

### 1 Sélection

Comme nous avons déjà précisé, plusieurs méthodes de sélection sont possibles, dont la roue biaisée, le *ranking* linéaire ou exponentiel, le K-tournoi... Les méthodes par *ranking* sont peu robustes et nécessitent de trier tous les chromosomes de la population selon leur *fitness*. Cette opération est coûteuse en temps de calcul.

Nous avons limité l'étude aux méthodes de sélection en K-tournois, en faisant varier le paramètre K.

Pour une population de taille  $N$ , la méthode peut être décrite par trois étapes.

1. Choisir  $K$  individus parmi les  $N$  présents dans la population.
2. Recopier le meilleur de ces  $K$  individus dans la nouvelle population.
3. Recommencer l'opération  $N$  fois.

Un paramètre  $K$  élevé a tendance à accroître la pression sélective, tandis qu'une valeur de  $K$  faible la diminue. Une forte pression sélective se traduit par une faible « espérance de vie » des chromosomes médiocres.

### 2. Croisement

Nous avons limité notre choix pour un croisement uniforme. Son avantage est de fournir des résultats très corrects sans pour autant être trop coûteux en temps de calcul.

### 3. Mutation

Le choix du taux de mutation a une grande influence sur l'efficacité de l'algorithme. Traditionnellement, pour les premiers essais la formule  $\mu = \frac{1}{l}$ , (où  $l$  est la longueur d'un chromosome) est fréquemment utilisée. Ce taux garantit qu'en moyenne, un gène par chromosome sera affecté.

## 7.1.5 Meilleur individu

C'est le chromosome dont l'énergie totale est minimale. Ce chromosome a des gènes récupérés à partir d'autres chromosomes.

**Conclusion**

Cette partie nous a permis de mieux comprendre l'intérêt de la méthode des contours actifs, pour la recherche d'une courbe de segmentation à une image

Les algorithmes génétiques ne rencontrent pas de problème concernant l'optimisation :

Il est vrai que des problèmes complexes peuvent donner des difficultés surtout pour la mise en équation de la fitness, mais une fois qu'elle est trouvée et que les opérateurs sont adaptés au problème, sa résolution peut avoir lieu.

# Chapitre 5

## Mise en œuvre

### 1. Introduction

En traitement d'image, la segmentation est une étape très importante qui requière beaucoup de calculs et du rapidité ; pour ces raisons, nous avons choisi d'utiliser le langage C++ qui se caractérise par la notion de la programmation orientée objet , la souplesse de la gestion et la manipulation de la mémoire grâce à l'allocation libre.

### 2. Interface de logiciel

L'interface de notre logiciel est découpée en deux grandes parties :

- L'interface segmentation d'images statiques ;
- L'interface de segmentation des séquences d'images (vidéo);

#### 2.1. Description de la première partie de l'interface

Cette partie est découpée en : un ensemble de menus, un ensemble d'onglets, un espace de travail et un espace de traitements.

##### 2.1.1. Les menus :

###### 2.1.1.1. Le menu Fichier :

Permet les différentes manipulations sur les fichiers (Images de format BITMAP) concernant :

- L'ouverture d'image ;
- La sauvegarde d'image ;
- L'impression et la fermeture des images de l'espace de traitement et de l'application.

###### 2.1.1.2. Le menu opération :

Permet de faire plusieurs opérations telle que :

- Conversion de l'image en niveau de gris ;
- Inversion des couleurs d'image ;
- image en noir et blanc ;
- inversion noir et blanc ;

###### 2.1.1.3. Le menu prétraitement :

Permet d'effectuer un filtrage sur l'image, il se décompose en trois sous-menus :

- filtrage (linéaire, non linéaire, morphologique) ;
- histogramme ;
- rehaussement de contraste.

#### 2.1.1.4. Le menu Segmentation :

Permet d'effectuer une segmentation d'image par quelques méthodes classiques.

#### 2.1.1.5. Le menu Aide :

Permet l'affichage d'aide de notre logiciel pour faciliter le travail à l'utilisateur.

### 2.1.2. Description de la première partie de l'interface (images statiques).

C'est le noyau de la segmentation d'images statiques, car il permet d'effectuer :

- Une segmentation selon trois méthodes : **Greedy** , **Greedy autonome** et algorithme génétique .
- **Une comparaison** des trois méthodes de segmentation.

#### 2.1.2.1. Le module Greedy

Ce module implémente les contours actifs par la méthode de segmentation de Greedy, avec toutes les opérations nécessaires, telles que l'initialisation du contour et l'affichage du contour .

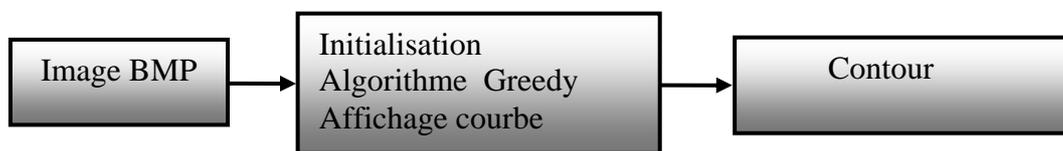


Figure 5.1 : Fonction du module Greedy

L'interface de paramétrage de l'algorithme de Greedy est donnée par la figure 5.2, elle résume tout ce qui peut être contrôlé par l'utilisateur, entre autres le choix des paramètres de pondération des cinq termes de l'équation (4.3), la manière de calculer les gradients, la condition de son arrêt, le nombre de voisins et la distance initiale entre les points au moment de l'initialisation.

L'initialisation du contour par un rectangle se fait par le dessin direct de ce dernier sur l'image. Notre module de segmentation est interactif, il est découpé essentiellement en deux parties : le réglage des paramètres et l'initialisation du contour.

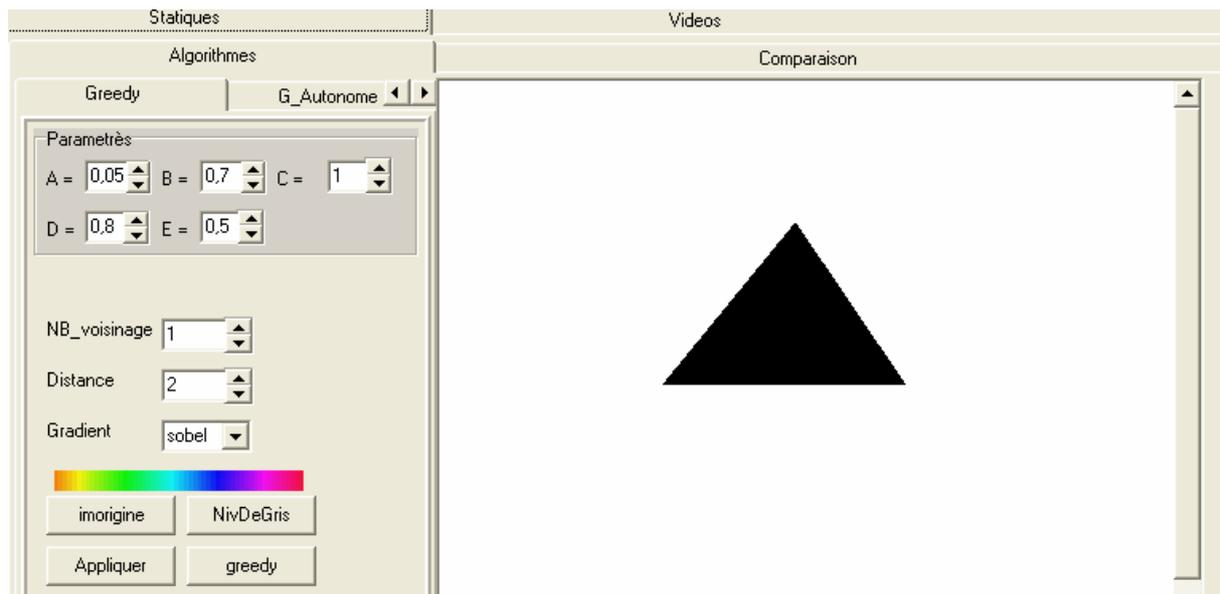


Figure 5.2 : Module de Greedy simple (image statique)

#### 2.1.2.1.1 Le réglage des paramètres :

Ce module permet le réglage des paramètres des contours actifs. Les paramètres sont la rigidité A, l'élasticité B, le coefficient de pondération du gradient d'image C, le coefficient de pondération d'intensité D et le coefficient de pondération de l'énergie de ballon E. Ainsi que d'autres paramètres comme le nombre de voisins, le nombre d'itérations et la distance initiale entre deux points adjacents.

#### 2.1.2.1.2. L'initialisation du contour :

Permet de récupérer les matrices de niveaux de gris à partir de l'image chargée dans l'espace de travail.

#### 2.1.2.1.3. L'espace de travail :

Représente la zone (image) où l'utilisateur peut charger une image et effectuer la segmentation.

#### 2.1.2.1.4. L'espace de traitement :

C'est la partie de l'interface où on obtient le résultat de l'application

### 2.1.2.2. Le module Greedy autonome.

Ce module en l'implémentation des contours actifs par la méthode de segmentation de Greedy autonome, avec toutes les opérations nécessaires, telles que l'initialisation du contour, l'affichage du contour et l'extraction de l'objet détecté.

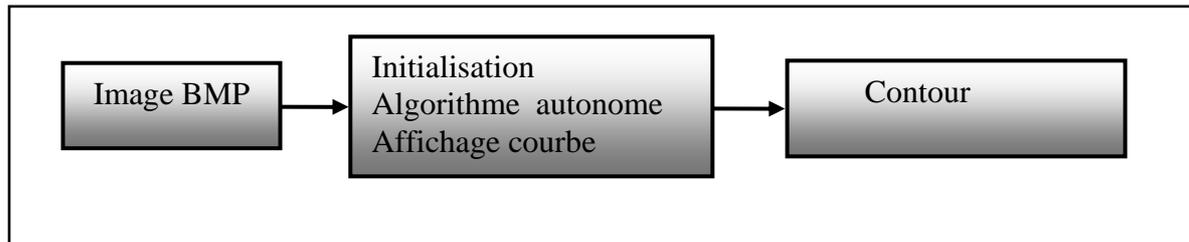


Figure 5.3 : Fonctions du module Greedy autonome

La démarche est la même que pour Greedy simple sauf qu'ici l'utilisateur ne peut pas contrôler les paramètres de pondération des cinq termes de la fonction d'énergie.

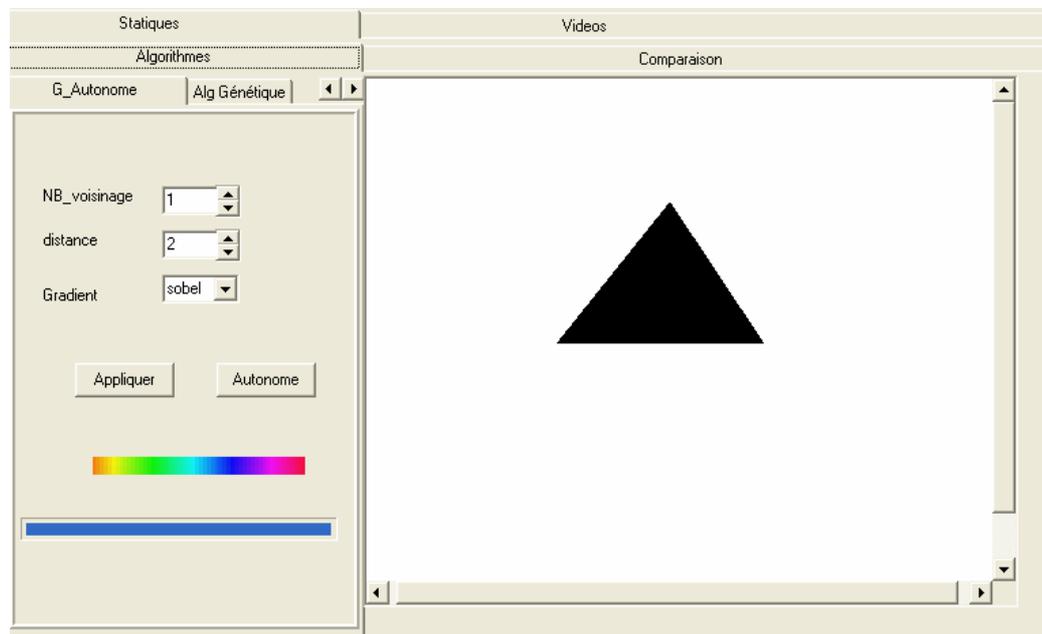


Figure 5.4 : Fonctions du module de Greedy autonome

### 2.1.2.3. Le module algorithme génétique

Ce module implémente les contours actifs génétiques, avec toutes les opérations nécessaires, telles que l'initialisation du contour, l'affichage du contour et l'extraction de l'objet détecté.

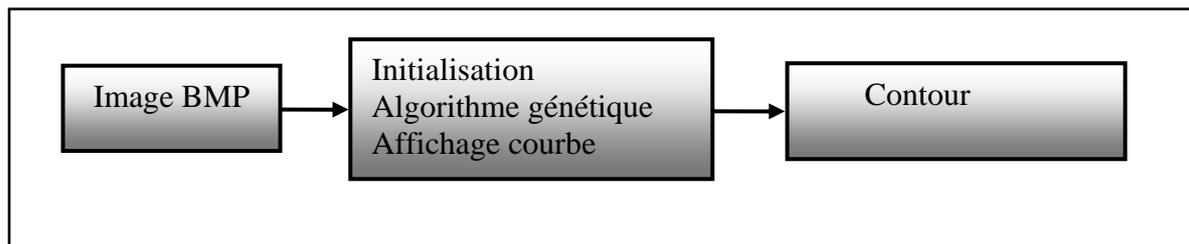


Figure 5.5 : Fonctions du module algorithme génétique.

Dans un premier temps on construit une population initiale, c'est-à-dire un ensemble de chromosomes qui représente des contours. Ces chromosomes doivent être déterminés aléatoirement en cliquant sur G Applique, c'est-à-dire pour chaque contour est choisi aléatoirement une série de points ; en initialisant le contour au centre de l'objet et en cliquant sur GApp, ces points vont subir des opérations de sélection, de croisement et de mutation pour converger vers un meilleur contour.

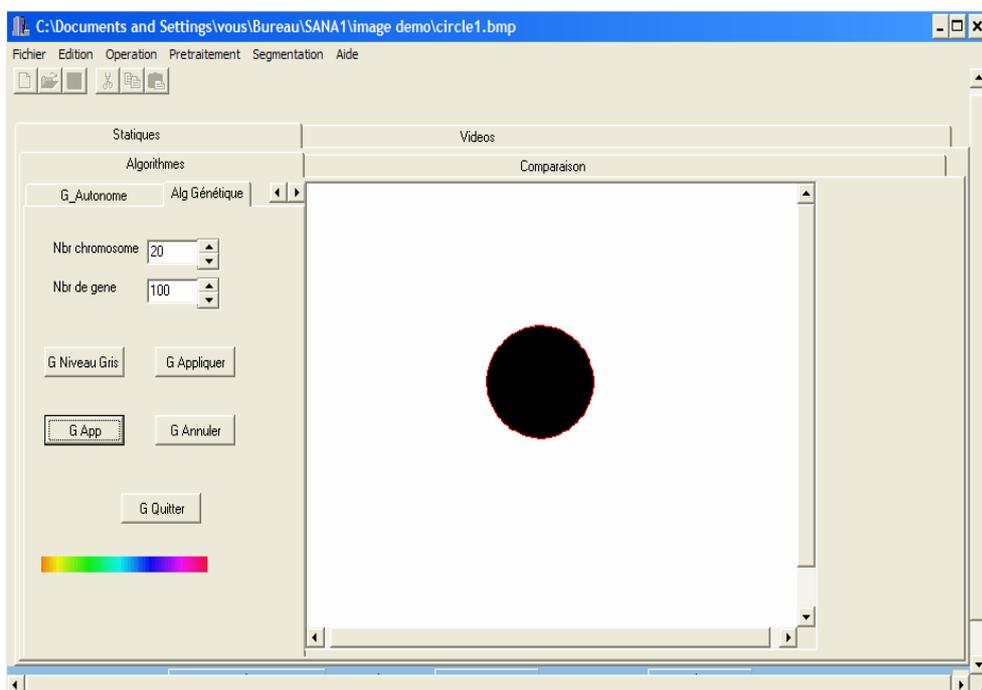


Figure 5.6: du module algorithme génétique.

#### 2.1.2.4. Le module de comparaison

Permet de visualiser le résultat de la comparaison entre les trois méthodes de segmentation en affichant le temps d'exécution et le nombre d'itérations.

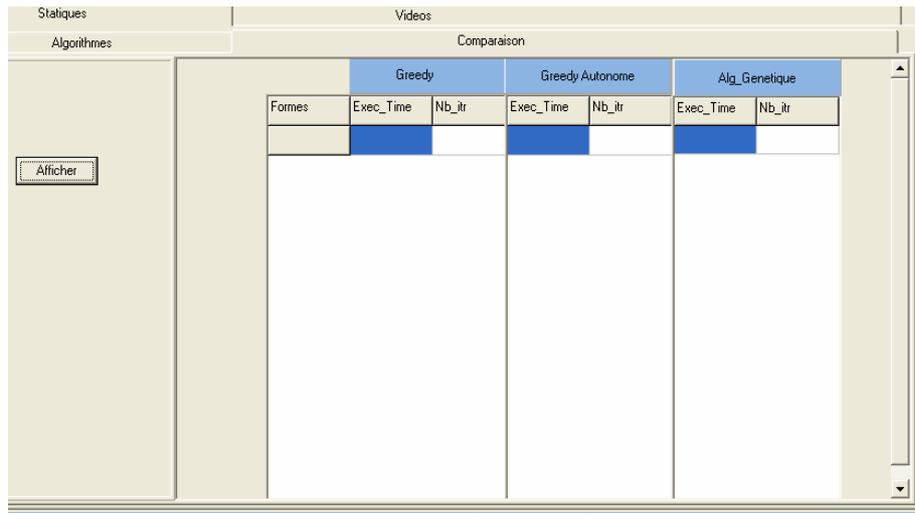


Figure 5.7 : Module de comparaison.

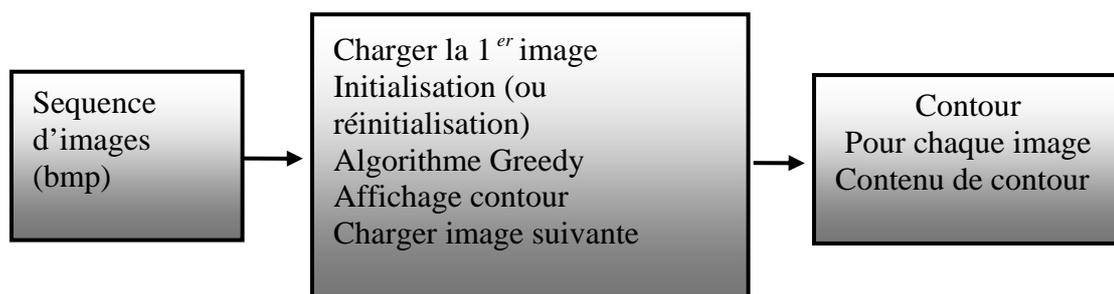
### 2.1.3. Description de la deuxième partie de l'interface (séquences d'images).

Cette partie est découpée essentiellement en deux parties :

- Un espace de travail ;
- Ainsi qu'un module pour la gestion de l'espace de travail.

#### 2.1.3.1. Pour le suivi par Greedy.

Ce module implémente le suivi des contours actifs par la méthode de segmentation de Greedy, avec toutes les opérations nécessaires, telles que l'initialisation et la réinitialisation du contour, l'affichage du contour, d'où l'extraction de l'objet détecté.



L'interface de paramétrage de suivi par l'algorithme de Greedy est donnée par la figure suivante :

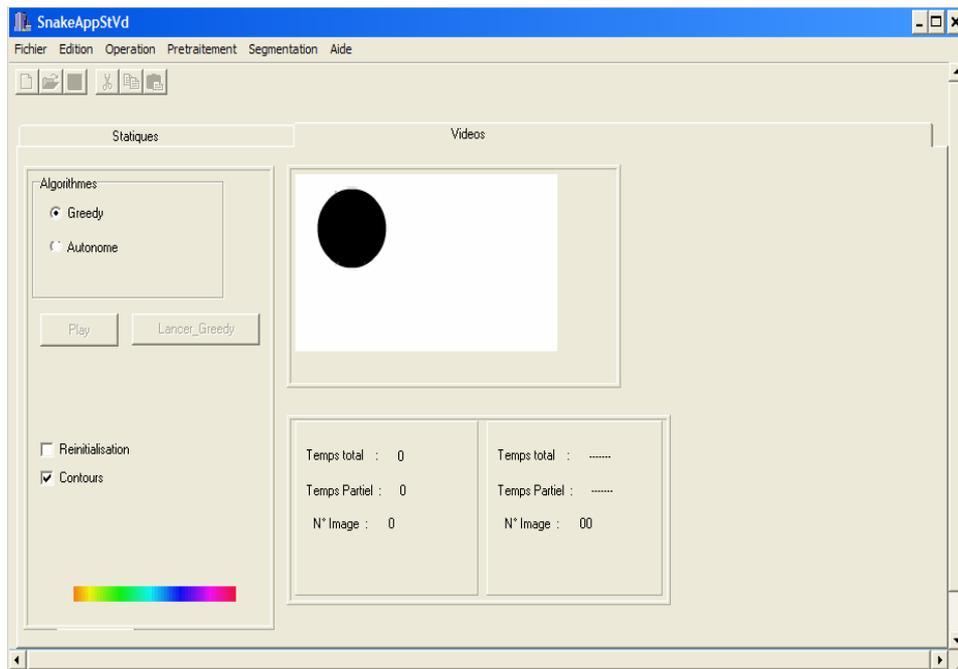
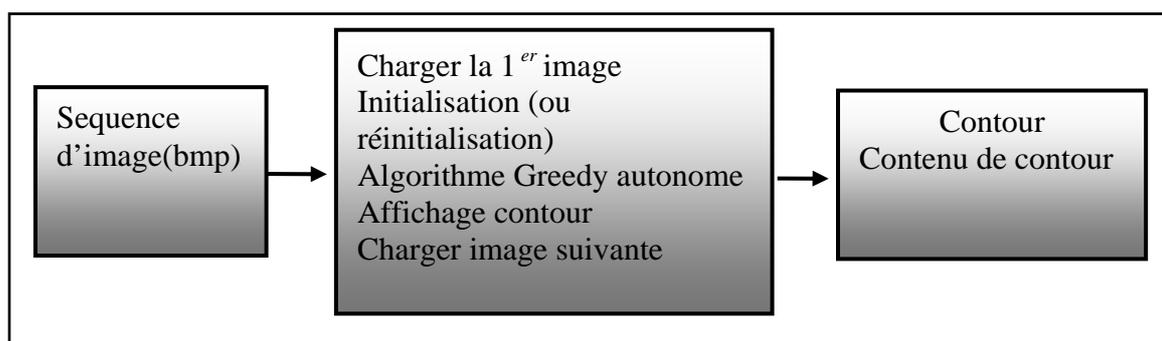


Figure 5.8 : module de suivi par l’algorithme de Greedy.

### 2.1.3.1. Pour le suivi par Greedy autonome.

Ce module implémente le suivi des contours actifs par la méthode de segmentation de Greedy autonome , avec toutes les opérations nécessaires , telles que l’initialisation et la réinitialisation du contour, l’affichage du contour et l’extraction d’objets détectés



L’interface de paramétrage de suivi par l’algorithme de Greedy autonome est donnée par la figure5.9 :

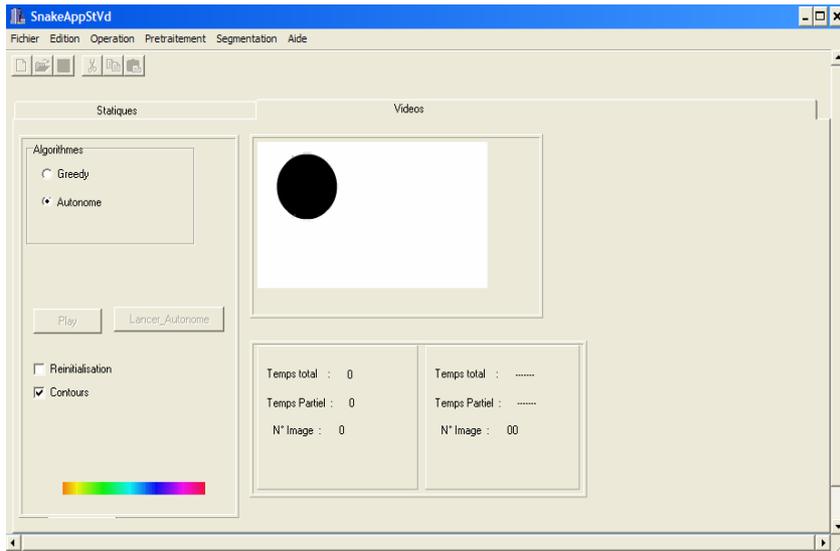


Figure 5.9 : module de suivi par l'algorithme de Greedy autonome.

### 3. Conclusion.

Au cours du présent chapitre, nous nous sommes intéressées à la structure de notre application et nous avons donné les principales routines et algorithmes composant notre logiciel. Dans le prochain chapitre, nous allons présenter une série de tests ainsi qu'une interprétation des résultats obtenus.

## Chapitre 6

### Tests et résultats

#### 1. Introduction

Dans ce chapitre, nous allons présenter les tests et résultats du travail fourni. Il s'agit principalement, de tester les trois algorithmes mis en œuvre dans le logiciel, à savoir : l'algorithme de Greedy simple et de Greedy autonome, que ce soit sur des images statistiques ou bien sur des séquences d'images ; puis l'algorithme génétique sur des images statiques. Ceci sera suivi d'une interprétation des résultats, pour pouvoir faire la comparaison entre les algorithmes utilisés.

Les tests ont été effectués sur un PC caractérisé par :

- Microprocesseur de 4.3GHZ
- RAM 224Mo
- Carte graphique intégrée de 32Mo

#### 2. Cas statique.

Pour Greedy simple les tests ont été effectués avec les paramètres suivants :

A = 0.05, B = 0.7, C = 1, D = 0.8 et E = 0.5

Comparaison						
Formes	Greedy		Greedy Autonome		Alg_Genetique	
	Exec_Time	Nb_itr	Exec_Time	Nb_itr	Exec_Time	Nb_itr
circle1.bmp	0:0:12	30	0:28:91	30	0:0:21	1
Nuage1.bmp	0:0:21	50	0:25:47	50	0:0:58	2
caree1.bmp	0:0:6	10	0:3:7	10	0:0:47	2

Figure 6.1 : Tableau de comparaison entre Greedy, Greedy autonome et l'algorithme génétique.

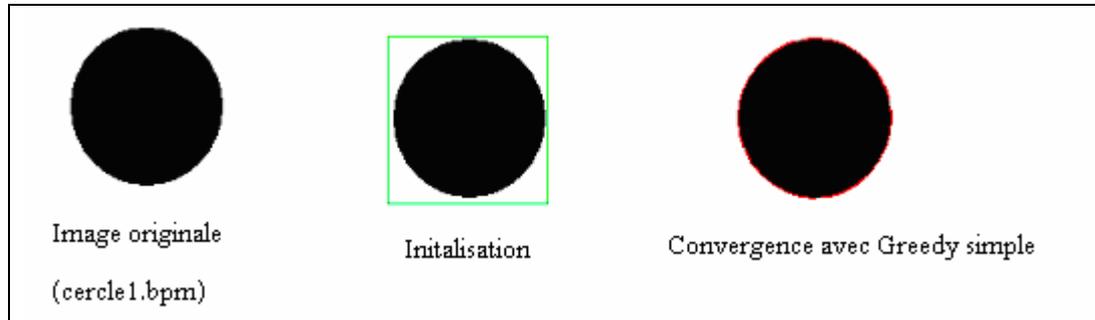
**a) Résultat pour l'image cercle.**

Figure 6.2 : Résultat obtenu par Greedy simple.

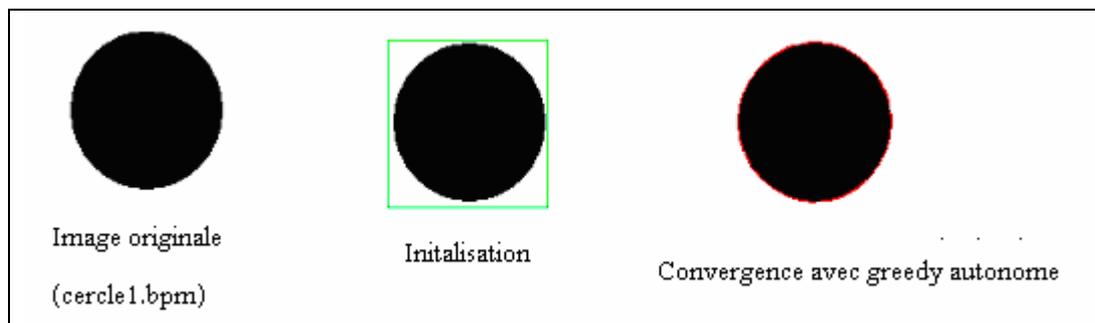


Figure 6.3: Résultat obtenu par Greedy autonome.

Pour le cercle, on constate que le contour initial converge totalement sur le contour de l'objet. Le temps de convergence est de 0,12 secondes pour Greedy et de 28,91 secondes pour Greedy autonome. Pour Greedy simple, les paramètres sont donnés directement par l'utilisateur, par contre pour Greedy autonome il y a un tirage aléatoire de paramètres locaux ; ce tirage vérifie le critère de convergence, ce qui nous a donné un temps d'exécution plus long.

Pour l'algorithme génétique, nous avons les résultats suivants:

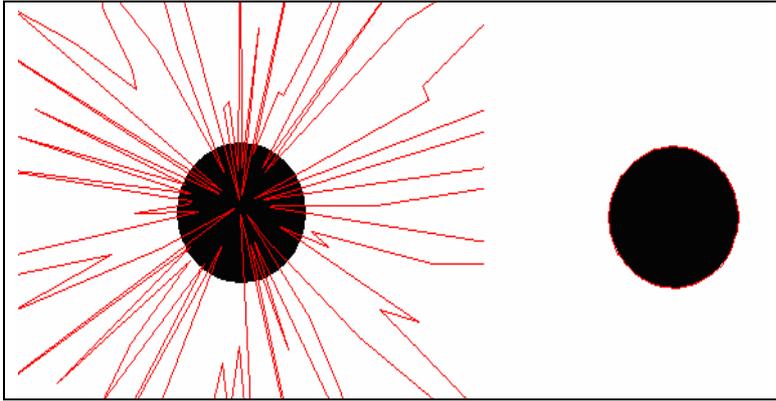


Figure 6.4: Résultat obtenu par l'algorithme génétique

Pour l'algorithme génétique, le contour initial converge sur l'objet, le temps de convergence est de 0,21 secondes. On constate que ce temps est supérieur au temps d'exécution de Greedy simple et inférieur à celui de Greedy autonome car la phase d'exécution des opérateurs génétiques est plus rapide que la phase du tirage aléatoire des paramètres.

En terme d'itérations, l'algorithme génétique a l'avantage de converger directement sur l'objet, après initialisation et exécution des opérateurs génétiques ; par contre pour Greedy et Greedy autonome l'algorithme passe par des contours intermédiaires avant convergence.

#### **b) Résultat pour l'image nuage.**

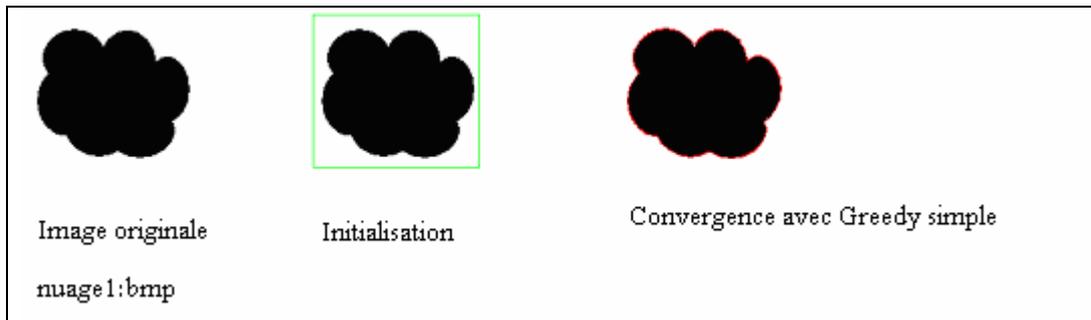


Figure 6.5 : Résultat obtenu par Greedy simple.

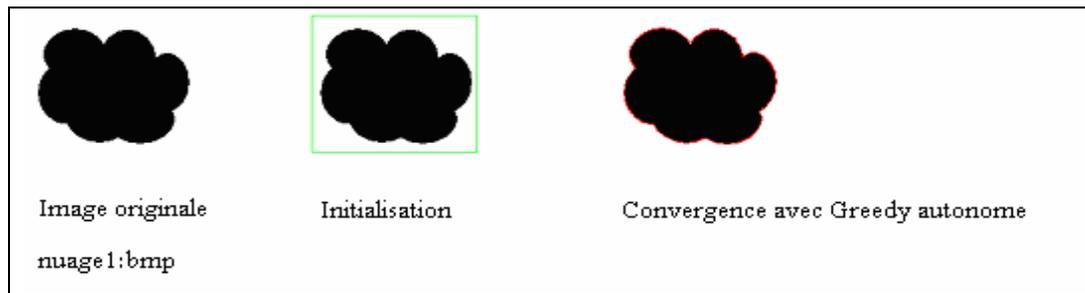


Figure 6.6: Résultat obtenu par Greedy autonome.

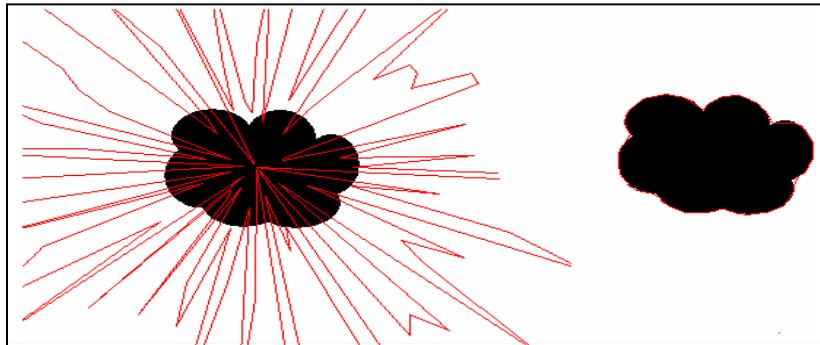


Figure 6.7 : Résultat obtenu par l'algorithme génétique.

Pour le nuage, on constate que le contour initial a une convergence totale sur le contour de l'objet pour les trois algorithmes.

Le temps de convergence est de 0,21 secondes pour Greedy et de 25,47 secondes pour Greedy autonome avec 50 itérations pour les deux algorithmes.

Pour l'algorithme génétique, le temps d'exécution est de 0,58 secondes avec 2 itérations. Les remarques décrites pour le cercle sont analogues pour le nuage.

### c) Résultat pour l'image carré.

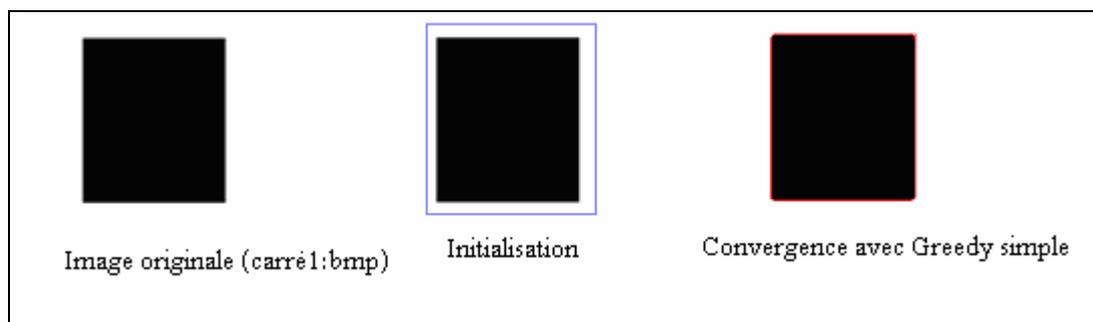


Figure 6.8: Résultat obtenu par Greedy simple

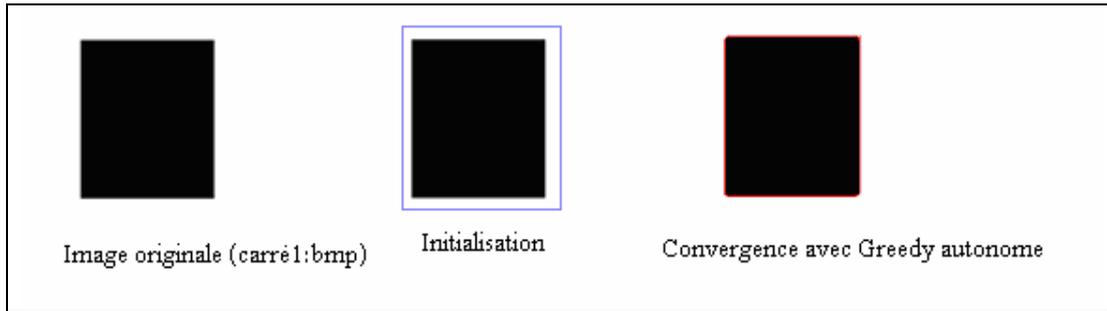


Figure 6.9 : Résultat obtenu par Greedy autonome

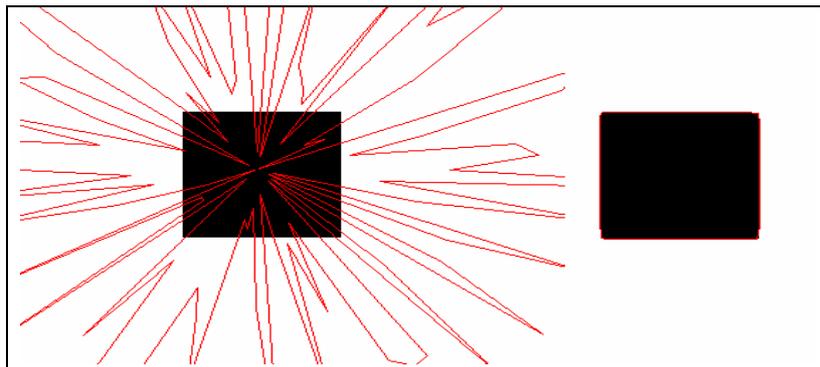


Figure 6.10 : Résultat obtenu par l’algorithme génétique.

Pour le carré, on constate que le contour initial a une convergence totale sur le contour de l’objet.

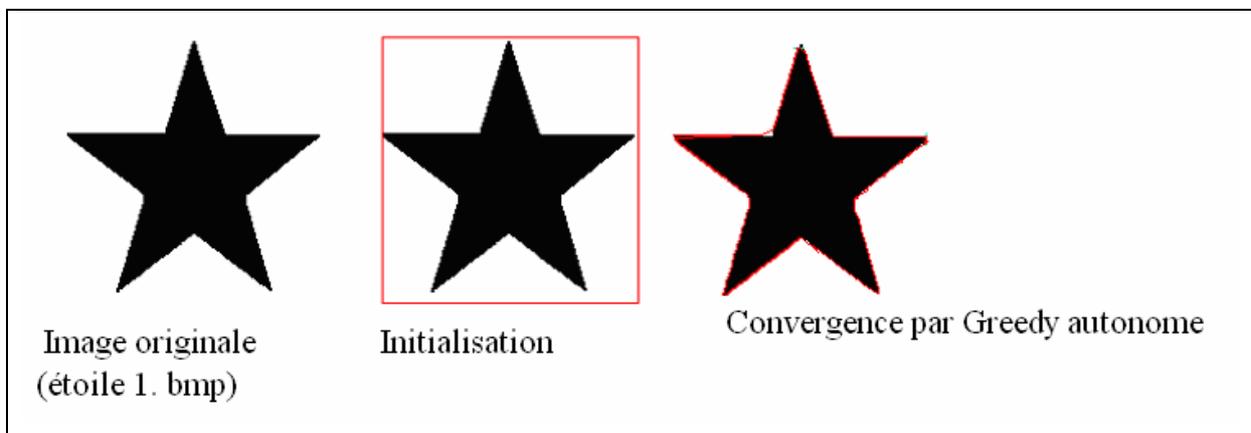
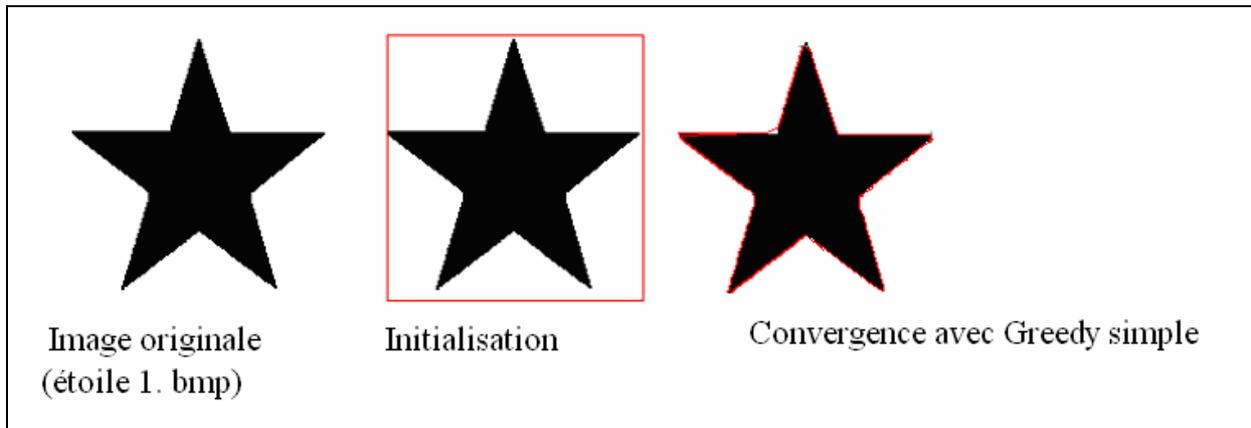
Le temps de convergence est de 0,6 secondes pour Greedy et de 3,7 secondes pour Greedy autonome avec 10 itérations pour les deux algorithmes.

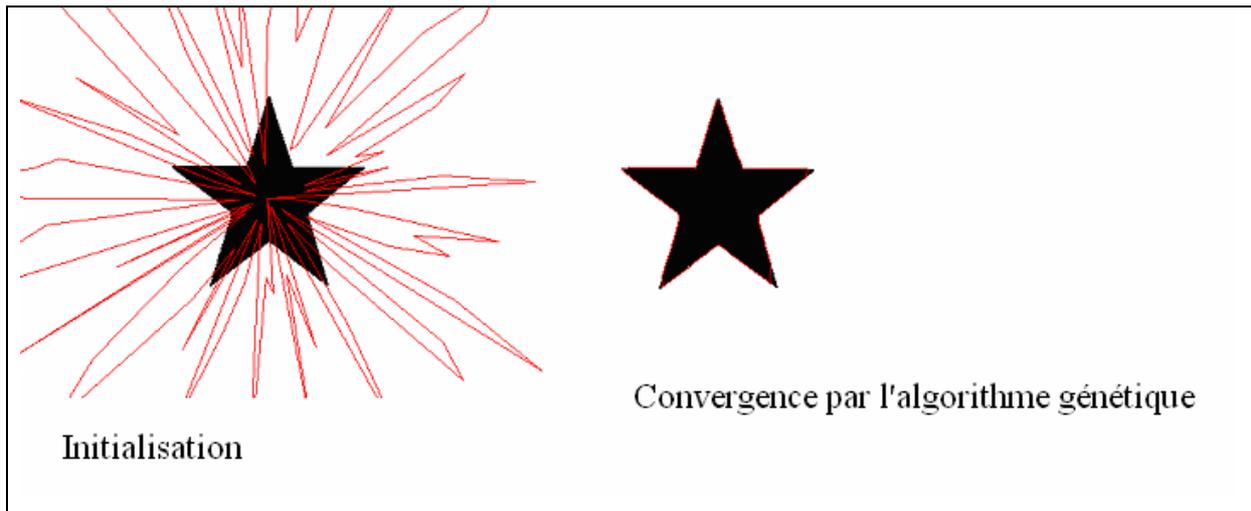
Pour l’algorithme génétique, le temps d’exécution est de 0,47 secondes avec 2 itérations. Les remarques décrites pour le cercle sont analogues pour le carré.

Pour confirmer l’efficacité de nos algorithmes nous les avons testés sur d’autres formes d’images, nous remarquons que les résultats sont satisfaisants.

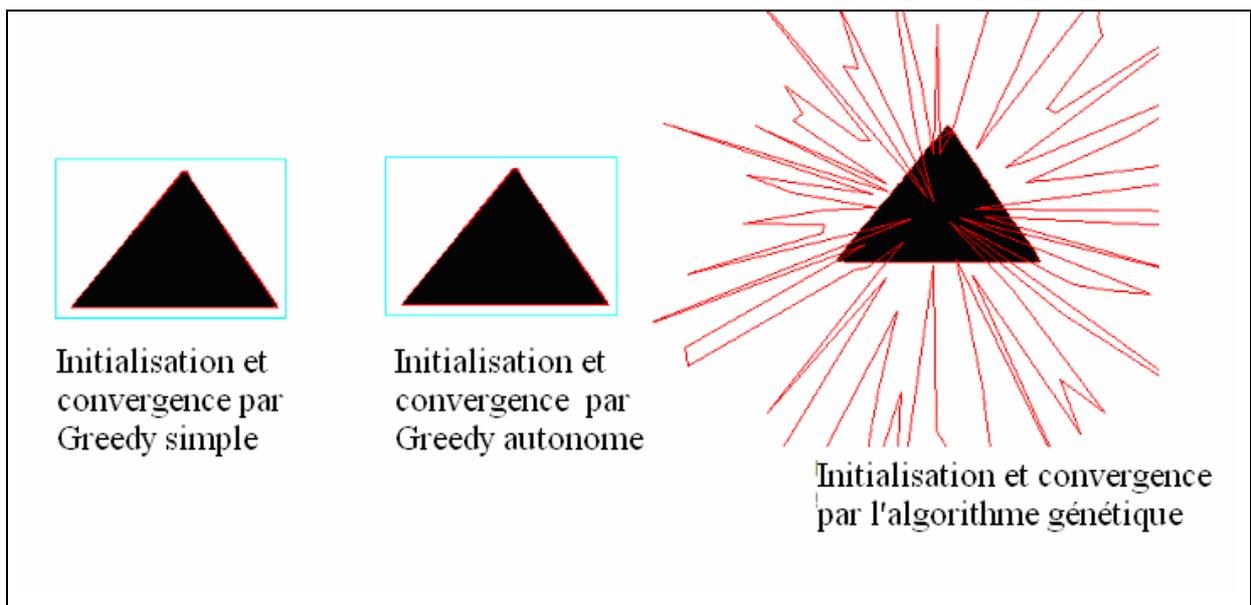
Nous citons par exemple : l’étoile et le triangle et nous donnons le contour final et le résultat de la comparaison, voir ci-dessous.

Comparaison						
Formes	Greedy		Greedy Autonome		Alg_Genetique	
	Exec_Time	Nb_itr	Exec_Time	Nb_itr	Exec_Time	Nb_itr
etoile2.bmp	0:0:29	64	2:15:4	64	0:0:19	1
trian1.bmp	0:0:14	52	1:13:14	58	0:0:25	1

**d) résultat pour l'image étoile**

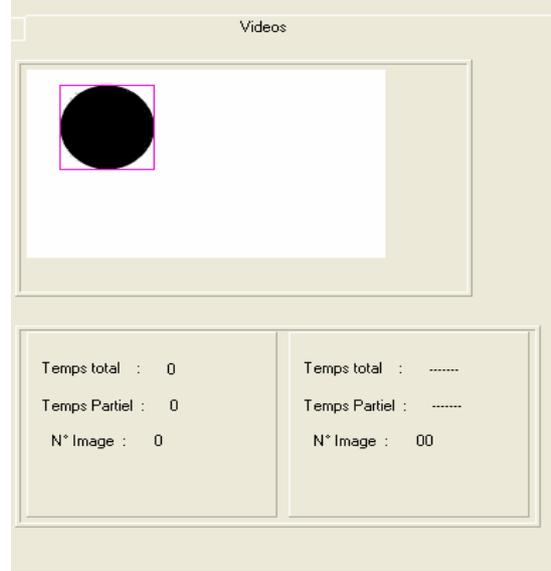


e) Résultat pour l'image triangle



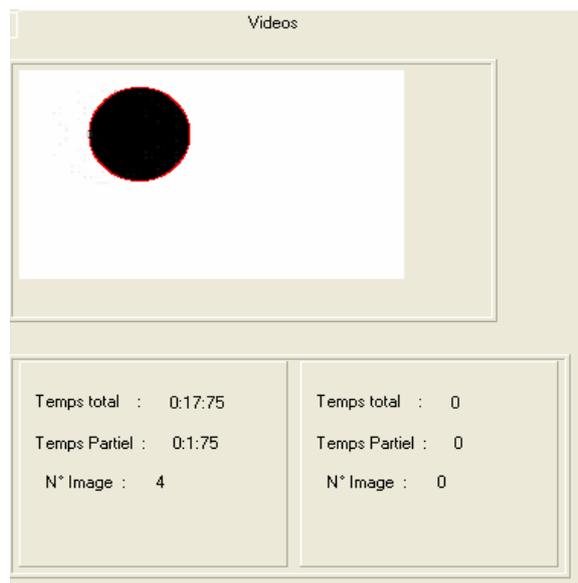
### 3. Cas dynamique (séquences d'images).

Pour des images en mouvement, nous avons pris une séquence d'images d'un cercle.



**Initialisation pour les deux algorithmes.**

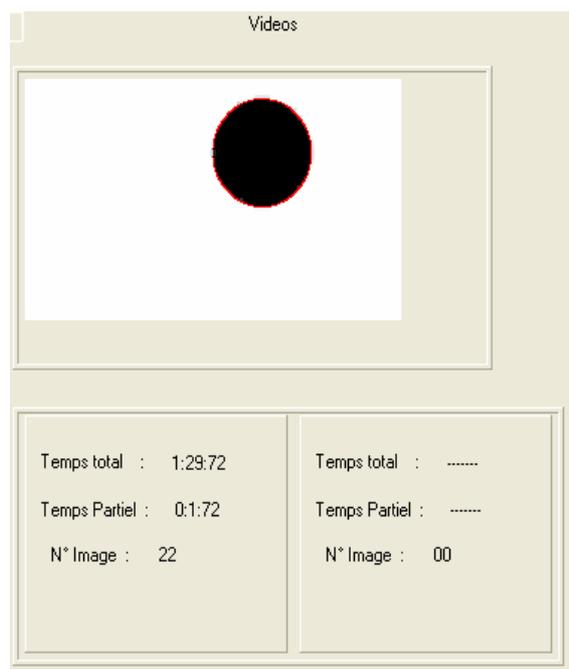
#### a) Résultat pour Greedy simple.



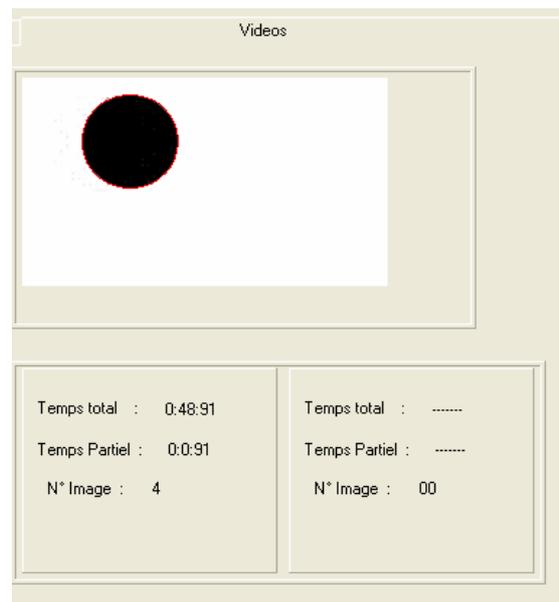
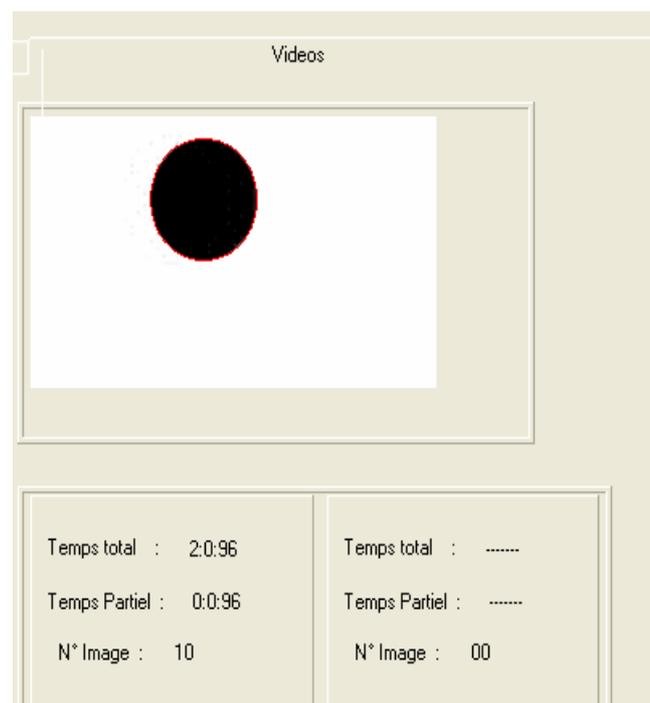
**Détection du contour de la quatrième image de séquence par l'algorithme Greedy simple.**

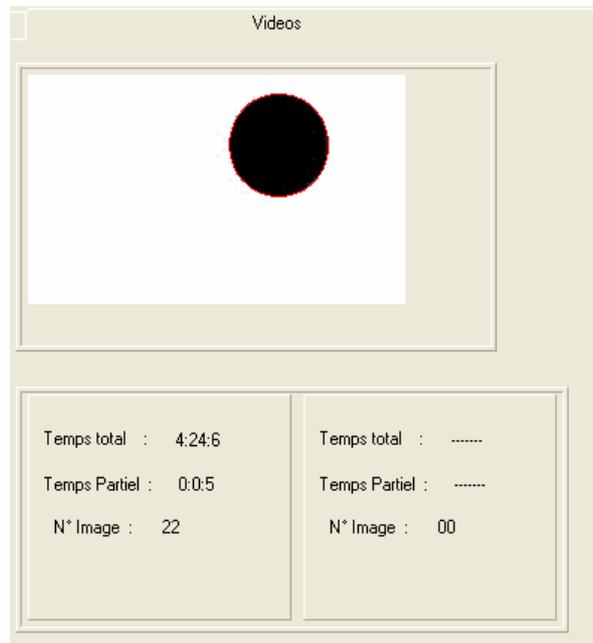


**Détection du contour de la dixième image de séquence par l'algorithme Greedy simple.**



**Convergence du contour de la dernière image par Greedy simple.**

**b) Résultat pour Greedy autonome.****Détection du contour de la quatrième image de séquence par l'algorithme de Greedy autonome****Détection du contour de la dixième image de séquence par l'algorithme de Greedy autonome.**



### **Convergence du contour de la dernière image par Greedy autonome.**

En terme de qualité de traitement, on constate que le contour est appréciable pour les deux algorithmes sur chaque image de la séquence ; le temps partiel entre les deux images successives de la séquence est de 2 secondes pour le suivi par Greedy simple et de 6 seconde pour Greedy autonome.

Pour la séquence avec 22 images, le temps du traitement est « 1 :29 :72 » pour Greedy et de « 4 :24 :6 » pour Greedy autonome.

#### **4. Conclusion.**

Comme nous l'avons remarqué dans le cas statique, le temps de traitement avec l'algorithme Greedy est inférieur au temps de traitement avec l'algorithme Greedy autonome, cela est dû aux tirages aléatoires des paramètres locaux choisis au début du traitement et à chaque image de la séquence. Pour les algorithmes génétiques, le temps de traitement dépend des opérateurs de reproduction.

# Conclusion générale

Le but essentiel de notre travail était l'étude des contours actifs et leur application dans le domaine du traitement d'images et plus exactement dans la segmentation.

Pour cela, nous avons commencé par un rappel sur la segmentation par les méthodes classiques à savoir l'approche contour et l'approche région. Les contours actifs ou snakes, faisant l'objet de recherche depuis plus de deux décennies, et présentant des variantes dérivant des approches classiques ont été aussi présentées. Intéressées plus particulièrement par l'approche contour, nous avons présenté quelques implémentations les plus connues à savoir l'approche variationnelle et l'algorithme de Greedy ; tous deux ayant pour but la minimisation d'une certaine énergie.

Dans ce même contexte de minimisation, nous avons proposé d'utiliser la puissance exploratoire des algorithmes génétiques. Ces derniers consistent à faire évoluer une population de snakes vers un individu présentant une énergie minimale.

Nous avons implémenté les algorithmes de Greedy et Greedy autonome, pour des images statiques et dynamiques. Ces algorithmes sont classés parmi les meilleures approches de minimisation, vue leur rapidité, leur précision et ainsi leur possibilité d'introduire des énergies de contexte comme l'énergie de ballon.

Les algorithmes génétiques ont été aussi implémentés, permettant de trouver, par la méthode d'évolution génétique, une solution optimale représentée par le dernier gagnant. Ces algorithmes sont simples et très performants dans la recherche d'une solution optimale plus pertinente sur des images statiques. Nous avons jugé nécessaire de comparer les trois algorithmes implémentés suivant leur temps d'exécution et le nombre d'itérations.

Dans le cas des images dynamiques la comparaison se fait uniquement par rapport au temps d'exécution.

Nous constatons une bonne qualité de traitement pour les trois algorithmes et un temps d'exécution relativement bon pour l'algorithme utilisé.

Ce travail pourrait être amélioré, en effectuant une segmentation simultanée de plusieurs objets dans une image ou encore une segmentation de la vidéo en temps réel.

# Bibliographie

[ALE, 97] Alexander D, Statistical Modeling of Color Data and Model Selection for Region Tracking, PhD, University College London, England, 1997, p.244

[AMI, 88] AMINI, A A.TEHRANI S and WEYMOUTH TE, Using dynamic programming for minimising the energy of active contours in the presence of hard constraints, 2<sup>nd</sup> Int. Conf. Comput.Vision, Tampa, FL, USA, décembre 1988, P. 95-99.

[ANW, 01] Alfred ANWANDER, Segmentation d'images couleur par un opérateur gradient Vectoriel multi- échelle et contour actif : Application a la quantification des phases Minéralogiques du clincker de Ciments , Thèse de doctorat, Ecole Doctorale des sciences de l'Ingénieur de Lyon, 2001.

[BED, 94] François BERARD , Vision par Ordinateur pour la Réalité Augmentée : Application au Bureau Numérique , DEA \_informatique, Ecole nationale polytechnique de Grenoble ENSIMAG – université Joseph Fourier, 1994.

[BIC, 02] Elodie BICHOT, Utilisation des contours actifs pour la segmentation vidéo, Projet DEA Image & calculs, 2002.

[BLO,05] I. BLOCH, Y. GOUSSEAU, H. MAITRE, D. MATIGNOU,B , Le traitement des images, Polycopiè du cours ANIM, Département TSI - Télécom-Paris, version 5.0, 2005.

[BOS, 00] Mattia BOSCO, Extraction d'une silhouette d'une séquence vidéo II, Laboratoire d'infographie (LIG) EPFL, 2000.

[CHR, 03] CHRISTOPHE Lége, Segmentation : approches contours/régions Traitement des Images, ESPEO, 2002-2003.

- [COC, 95] J.P.COCQUIREZ et s PHILIPP, Analyse d'images filtres et segmentation, Edition Masson, Paris, 1995.
- [COH, 89] L.D.COHEN, On active contour models.rapport de Recherche 1075, INRA, August 1989.
- [COH, 91] L.Cohen, On active contour models and balloons. CVGIP: Image Understanding,53(2):211-218,1991.
- [COU, 05] Antoine COUTANT, La méthode des contours actifs en traitement d'images, Mémoire pour l'examen probatoire en Calcul Scientifique, Conservatoire Nationale des Arts et des métiers, Paris, 2005
- [DAV, 89] C.David and S.W.Zucker, Potential,Valleys, and Dynamic Global Coverings. Technical report, McGill University, 1989.
- [FOR, 03] Guillaume FORET, Segmentation spatio-temporelle d'objets vidéo en vue de leur caractérisation, Thèse de doctorat, Institut national Polytechnique de Grenoble 2003.
- [HOR, 74] S.L, HOROWITZ , T. PAVLIDIS, Picture Segmentation Split And Merge Procedure, proceeding of the international joint conference on pattern Recognition, Août 1974, P424-433
- [HUA, 03] HUART Jérémy, Segmentation Locale Pour l'extraction d'Objets dans les Images et les Séquences Vidéos, Rapport de DEA, INPG , 2003.
- [KAS,88] KASS, M. WITKIN, A. and Terzopoulos D, Snake active contour models, International Journal of Computer Vision, 1:321-332, January 1988.
- [KHE, 01] KHEFFACHE Madjid et SLIMANE Ramdane , Segmentation d'images médicale avec contours actifs , Mémoire d'ingénieur, INI 2001.

[LAS,04] N. LASSOUAOUI, Segmentation des images par différentes approches et optimisation avec les algorithmes génétiques, Thèse de doctorat en électronique, ENP, 2004.

[LEF, 01] Sébastien LEFEVRE, Cyril FLUCK, Benjamin MAILLARD , Nicole VICENT, Un modèle de contour actif pour le suivi rapide d'objet en mouvement, Application au suivi de joueurs de football ,publication de l'équipe de RFAI : colloque GR ETSI sur le traitement de signal et des images, Toulouse (France), Septembre 2001.

[LEJ, 01] Fabian LE JEUNE, Suivi de la Main dans une Séquence Vidéo, Stage de DEA Université de Marne la Vallée 2001.

[MEN, 90] A. Menet, P. Saint-Marc, and G. Medioni, Active contour models: Overview, implementation, and applications, In IEEE Conf. Syst. Man. Cyb. L.A. Nov 90.

[PAP, 99] Crisophe PAPIN, Analyse spatio-temporelle d'Image satellitaires Météologiques : détection et suivi de structures nuageuses critiques , Thèse de doctorat Université de Rennes 1, 1999.

[RAD, 95] Radeva , P, Marti, E, Facial Features Segmentation by Model Based Snakes, Conference Trolbada de Joves Investigators, Bellaterra (bercelona), Espagne, Septembre 1995.

[ROU, 03] J-J ROUSSELLE , Les contours Actifs, une méthode de segmentation : Application à l'imagerie médicale, Thèse de doctorat, Université de Tours, 2003.

[SHA,92] D-J WILLIAMS, M. SHAH, A fast Algorithm for active contours and curvature estimation, CVIGP Computer Vision Graphics Image Process, Image Understanding Vol 55, N°1 janvier 1992, P 14-26

[VAL, 01] Sébastien VALLETTE, Isabelle MAGNIN et Rémy PROST, Segmentation vidéo et suivi d'objets par maillage déformable : gestion des discontinuités de mouvement, CREATIS, UMR CNRS 5515, affilié à l'INSERM, INSA, bâtiment blaise pascal, Villeurbanne, France, 2001.

[ZUC, 88] S.W.Zucker,C.David, A.Dobbins, and L.Iverson.The Organization of Curve Detection: Coarse tangent fields and fine spline coverings. Technical report, McGill University, 1988.