

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERY, Tizi-Ouzou**



**Faculté de Génie Electrique et d'Informatique**  
**Département d'Automatique**

**Mémoire de Fin d'Etudes**

En vue de l'obtention du diplôme

*Ingénieur d'état en automatique*

***Thème***

**Etude d'un contour actif  
rapide**

Dirigé par :  
- M<sup>elle</sup> Ouardia Chilali

Présenté par :  
- Kamel Bellahcene

Soutenu le : 16/09/2012

*Promotion 2012*

Ce travail a été préparé à : Université Mouloud MAMMERY, Tizi-Ouzou

# Remerciements

*Je tiens à remercier particulièrement M<sup>lle</sup> CHILALI OUARDIA, qui m'a proposés d'effectuer un projet de fin d'étude, pour lequel elle a fait montrer d'une disponibilité de tous les instants et qui m'a fait profiter de son large expérience tout au long de la réalisation de ce projet.*

*Je remercie les membres de jury, qui nous feront l'honneur de participer au jugement de ce travail.*

*Par ailleurs, mes remerciements vont à l'ensemble de nos professeurs du génie Electrique, qu'ils veuillent bien trouver ici l'expression de nes sentiments les plus respectueux.*

*Un grand merci à toute ma famille, plus particulièrement à mes parents, pour m'avoir soutenu et aidé tout au long de mes études.*

*Je tiens aussi à remercier vivement tous ceux qui m'ont aidé de près ou de loin à élaborer ce projet. À savoir les collègues de ma promotion et tout mes amis.*

# *Dédicaces*

Je dédie ce modeste travail à :  
Mes chers parents, Mes frères et sœurs,  
Toute ma famille,  
Tous mes amis.

<b>Introduction générale.....</b>	<b>1</b>
<b>I Les contours actifs</b>	
I.1 Introduction.....	2
I.2 Définition.....	2
I.3 Principe des contours actifs.....	2
I.4 Représentation d'un contour actif.....	3
I.4.1 Approche implicite : les ensembles de niveaux.....	3
I.4.2. Approche explicite : les modèles paramétriques.....	5
I.5 Les approches des contours actifs selon l'énergie externe.....	6
I.5.1 Les contours actifs basés contours.....	6
I.5.1.1 Approches variationnelles.....	6
I.5.1.2 Approche non variationnelle.....	7
I.5.2 Contours actifs basés régions.....	8
I.6 Les méthodes d'accélération.....	9
I.6.1 Fast Marching.....	9
I.6.2 La bande étroite.....	11
I.6.3 Algorithme Hermes.....	12
I.6.4 Implémentation rapide de Shi et Karl.....	13
II.7 Conclusion.....	14
<b>II Approche d'accélération adoptée</b>	
II.1. Introduction.....	15
II.2. Approches développées.....	15

## *Sommaire*

---

II.3. Approche adoptée .....	17
II.3.1 La fonction distance signée discrète.....	17
II.3.2 Technique de l'évolution .....	19
II.3.3 Critère d'arrêt.....	21
II.4 les différentes forces externes.....	21
II. 4.1 la moyenne d'intensité .....	22
II.4.2 La moyenne et la variance.....	22
II.5 Extension aux trois dimensions .....	23
II.6 complexité algorithmique du modèle.....	23
II.7 Conclusion .....	26

### **III Tests et résultats**

III.1 Introduction.....	27
III.2 Matériel utilisé et présentation des images .....	27
III.2.1 Matériel utilisé .....	27
III.2.2 Présentation des images utilisées .....	27
III.3 Tests et résultats .....	29
III.3.1 Tester le contour initial.....	29
III.3.2 Application sur des images différentes .....	34
III.3.2.1 Application sur des images en niveaux de gris.....	34
III.3.2.2 Application sur des images en couleurs.....	35
III.3.2.3 Application sur des images bruitées.....	36
III.3.2.4 Application sur des images 3D .....	37

## *Sommaire*

---

III.4 Evaluation comparative.....39

III.5 conclusion.....44

**Conclusion générale**.....45

**Bibliographie**

# Introduction générale

---

La recherche automatique d'informations sur l'image s'avère un besoin. L'avancée technologique et l'explosion de l'informatique dans les années 80 a permis le développement du traitement d'images. La première étape dans le processus de compréhension d'une image est le plus souvent l'extraction des objets qui la composent. C'est ce qu'on appelle la segmentation des images. Plusieurs méthodes de segmentation ont vu le jour et existent abondamment dans la littérature.

Nous nous sommes intéressés à la méthode des contours actifs : une technique particulière de segmentation. Les contours actifs tirent leur origine des modèles élastiques, mais la communauté s'accorde à les attribuer à l'équipe Kass, Witkin et Terzopoulos, qui introduisirent les courbes minimisantes ou *snakes*. Depuis, les domaines d'utilisation sont nombreux tant en 2D qu'en 3D, et de nombreuses versions sont développées.

Cependant, cette méthode de segmentation, à l'image des autres méthodes, présente des insuffisances auxquels il faut remédier. Dans les contours actifs géométriques, le temps de calcul reste un problème qu'il faut résoudre surtout lorsque les données sont très volumineuses. En effet, les algorithmes mis en œuvre nécessitent toujours de plus en plus de calcul car le volume des images elles-mêmes augmente sans cesse. Plusieurs techniques ont été proposées pour remédier à cet inconvénient.

Pour cela, l'objectif de ce travail est d'étudier un contour actif rapide, tout en assurant une bonne segmentation. Pour se faire, nous avons organisé notre travail comme suit :

Dans le **chapitre 1**, nous nous intéresserons aux modèles des contours actifs d'une manière générale, tout en exposant leurs principes, les différents types et les méthodes d'accélération.

Dans le **chapitre 2**, nous présenterons le modèle de l'approche adoptée tout en expliquant son principe, ainsi que l'algorithme d'implémentation.

Le **chapitre 3** sera consacré aux différents tests effectués sur différentes images. Un test de comparaison de l'approche avec d'autres est exposé dans ce chapitre.

Le mémoire s'achèvera par une conclusion générale.

## I.1 Introduction

Les contours actifs tirent leur origine des modèles élastiques, mais la communauté s'accorde à les attribuer à l'équipe Kass, Witkin et Terzopoulos, qui introduisirent les courbes minimisantes ou *snakes*. Depuis la publication de cette équipe, plusieurs modèles déformables ont vu le jour. De très nombreuses équipes s'y sont intéressées de manière plus ou moins approfondie. Les domaines d'utilisation sont nombreux tant en 2D qu'en 3D tels : la reconnaissance de formes, le suivi de scènes, la segmentation d'images.

Dans ce chapitre on s'intéressera au modèle du contour actif d'une manière plus avancée et plus détaillée tout en exposant son principe, ses différents types et, plus particulièrement, les méthodes d'accélération de son évolution.

## I.2 Définition

Un modèle de contour actif est formé d'une série de points mobiles et répartis sur une courbe en deux dimensions. La courbe (qui peut être fermée ou pas) se déplace et épouse les contours des objets en fonction de divers paramètres. Cette dynamique est basée sur la notion d'énergie interne, externe et de contexte

Le but étant de minimiser l'énergie totale présente le long de la courbe.

## I.3 Principe des contours actifs

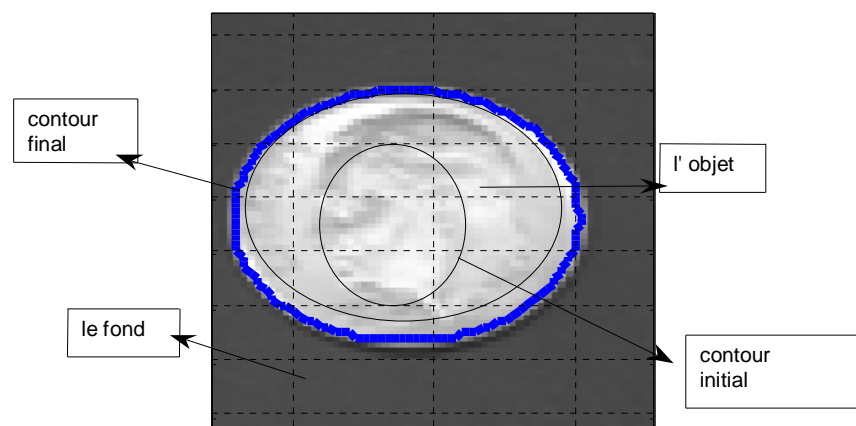


Fig. I.2. Evolution d'un contour actif

## I.4 Représentation d'un contour actif

Deux approches principales existent pour la mise en œuvre des contours actifs : les approches explicites et les approches implicites.

### I.4.1 Approche implicite : les ensembles de niveaux

Une approche dite implicite d'implémentation des contours actifs. L'idée est simple. Une courbe, qui borne une région ouverte, peut être définie à l'instant  $t$  au point  $s$  comme le niveau zéro d'une fonction  $\Phi$ .

$$X \in \Gamma(s, t) \leftrightarrow \Phi(X, t) = 0 \quad (\text{I.2})$$

Nous pouvons choisir  $\Phi$  comme étant une fonction distance signée au contour  $\Gamma(s, t)$  [4], négative à l'intérieur de la courbe et positive à l'extérieur (figure I.3).

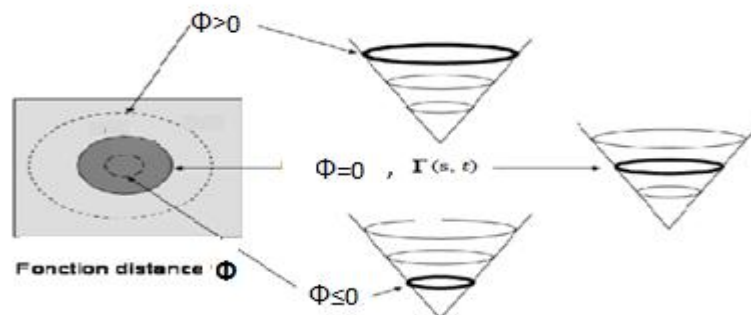


Fig. I.3. Principe des ensembles de niveaux.

l'équation d'évolution du contour actif est la suivante [3] :

$$\frac{\partial \Gamma(s, t)}{\partial t} = F(s, t) N(s, t) \quad (\text{I.5})$$

Avec  $F$  la vitesse d'évolution du contour actif et  $N$  la direction normale au contour.

Les avantages d'une telle méthode sont les suivants [5]:

- Les changements de topologie sont gérés automatiquement.
- La fonction distance  $\Phi$  permet des schémas numériques stables et précis.

- Les propriétés géométriques du contour peuvent être facilement estimées à partir de la fonction  $\Phi$ .
- La formulation peut être aisément étendue aux dimensions supérieures.

Cependant, cette approche reste tout de même très lourde en calculs, et ne permet pas de convertir facilement le contour en courbe géométrique orientée [4]. D'autres approches ont été développées, notamment les modèles paramétriques de contours actifs, permettant de limiter les temps de calcul.

#### **I.4.2. Approche explicite : les modèles paramétriques**

Les modèles paramétriques définissent le contour actif à partir d'un ensemble de paramètres. Les approches paramétriques ont l'avantage principal de réduire considérablement le temps d'exécution. Cependant, les changements de topologie ne sont plus gérés automatiquement et leur extension aux dimensions supérieures demeure difficile [4].

### **I.5 Les approches des contours actifs selon l'énergie externe**

#### **I.5.1 Les contours actifs basés contours**

Ils ne tiennent en compte que de l'information présente sur les bords des objets à segmenter. On pourra distinguer les modèles utilisant les méthodes variationnelles et ceux utilisant les méthodes non variationnelles [4]

#### **I.5.2 Contours actifs basés régions**

. Ces modèles présentent beaucoup de qualités, notamment, la capacité de gérer automatiquement les changements de topologie et l'insensibilité par rapport à l'initialisation du contour. Cependant, malgré toutes ces qualités, cette approche reste très couteuse en temps de calculs. Pour cela, plusieurs méthodes d'accélération sont introduites et exploitées.

### **I.6 Les méthodes d'accélération**

#### **I.6.1 Fast Marching**

L'évolution Fast Marching proposée dans [32 ; 1] a été développée dans le cadre de front se propageant de manière monotone. Ainsi, la fonction vitesse  $F$  (équation 1.5) permettant de déformer la courbe est soit positive (correspondant à une dilatation de la courbe), soit négative (correspondant à un rétrécissement), mais ne changera jamais de signe.

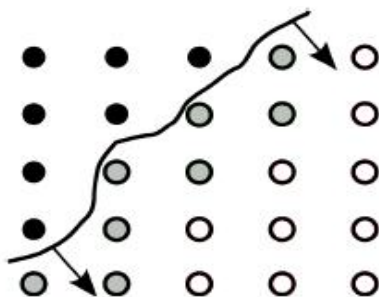


Fig.1.4. Différentes classes de pixels pour l'algorithme Fast Marching.

L'évolution du temps de passage du front en chaque pixel  $x$  de l'image est donnée par :

$$\nabla T(x) \cdot F(x) = 1 \text{ avec } T = 0 \forall x \in C(s) \quad (\text{I.14})$$

La fonction vitesse  $F$  étant considérée positive, l'algorithme de propagation est constitué d'une phase d'initialisation et d'une phase d'évolution (Algo. I.1).

L'algorithme s'arrête lorsque tous les points de l'image sont calculés.

Contrairement à l'implémentation classique en ensembles de niveaux, la méthode Fast Marching ne donne pas immédiatement les contours de l'objet recherché.

### I.6.2 La bande étroite

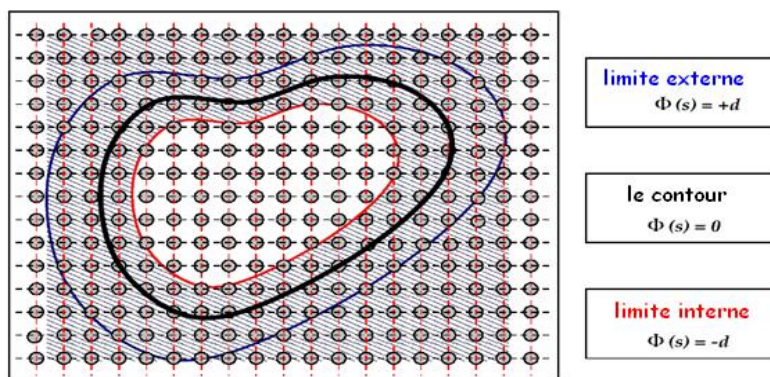
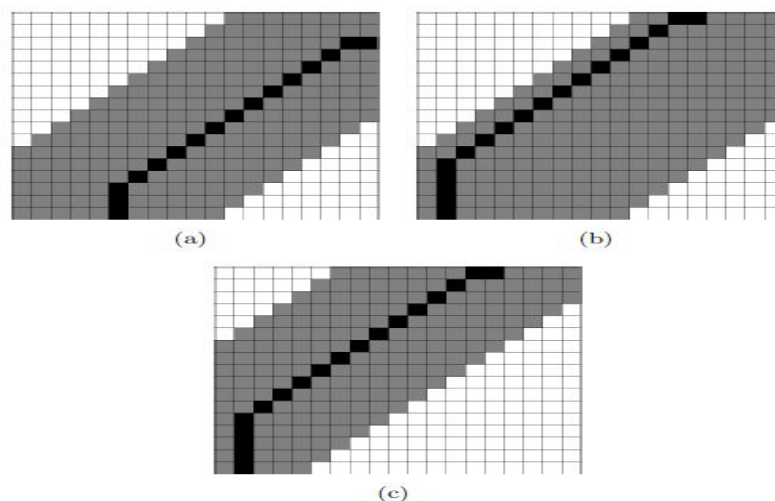


Fig. I.5. La bande étroite.

. La figure (I.6) illustre le principe de l'approche.



**Fig. I.6.** Principe de l'implémentation des ensembles de niveaux par bande étroite. Les pixels gris représentent la bande étroite et les pixels noirs le contour actif. a) la mise à jour de  $n$  n'est effectuée qu'à l'intérieur de la bande ; b) le contour est déplacé et se rapproche de l'extrémité de la bande ; c) la bande étroite est alors réinitialisée.

Cette implémentation reste plus lente que l'évolution Fast Marching mais, alors que cette dernière requière un front à propagation monotone, l'approche par bande étroite permet de gérer tout type de propagation de front [9].

Plus récemment, une amélioration de l'implémentation en bande étroite appelée Fast Narrow Band a été proposée dans [28]. Le principe est de calculer la bande étroite en utilisant des voisinages circulaires autour des pixels appartenant au niveau zéro de  $\phi$ .

### I.6.3 Algorithme Hermes

### I.6.4 Implémentation rapide de Shi et Karl

Au lieu d'être calculée à partir d'une distance, variant de manière continue dans  $\mathbb{R}$ , la fonction d'ensembles de niveaux ne peut prendre que quatre valeurs, qui indiquent le statut du point  $x$  :

$$\phi(x) = \begin{cases} -1 & \text{si } x \in L_{out} \\ 1 & \text{si } x \in L_{in} \\ -3 & \text{si } x \in R_{out} \text{ et } x \notin L_{out} \\ 3 & \text{si } x \in R_{in} \text{ et } x \notin L_{in} \end{cases} \quad (\text{I.15})$$

Où  $L_{in}$  et  $L_{out}$  sont les ensembles de pixels appartenant au front, respectivement à l'intérieur et à l'extérieur de la région.

## I.7 Conclusion

Nous avons présenté les différentes approches pour les contours actifs, de l'approche purement "contours" aux approches plus récentes qui intègrent des termes de régions dans les fonctionnelles à minimiser. Les avantages et les inconvénients sont relatifs aux types d'applications à réaliser. Seulement, nous nous sommes penchés sur les contours actifs implicites et nous avons développé certaines méthodes qui permettent d'accélérer la convergence de ce dernier.

Dans le chapitre suivant, nous allons développer une autre méthode d'accélération afin de la comparer à celles déjà étudiées.

## II.1 Introduction

La méthode *level set* est une technique couramment utilisée dans les applications de traitement d'image. En formulant le problème selon une fonctionnelle énergétique, la solution est, généralement, trouvée par descente de gradient. Toutefois, la méthode, en ce qui concerne le temps de calcul, reste très lente surtout lorsque les données sont très volumineuses.

Initialement, on a procédé à réduire le domaine du calcul à une zone entourant, immédiatement, le niveau zéro de la fonction *level set*. Depuis, plusieurs techniques ont été proposées pour remédier à cet inconvénient.

Dans ce chapitre, nous allons détailler une de ces techniques, en présentant son principe. Seulement, nous allons faire un petit rappel sur les techniques déjà développées dans notre université.

## II.2 Approches développées

Il y a lieu de dire que plusieurs méthodes d'accélération ont été développées dans la littérature (section I.6). En effet, nous avons parlé de la méthode dite de la bande étroite. Elle consiste à ne faire les calculs que sur une bande entourant le niveau zéro de  $\Phi$  et à remettre à jour cette bande à chaque fois que la courbe s'approche de ses bords. Ainsi, les coûts de calcul sont nettement réduits. Nous avons, aussi, cité le Fast marching et l'algorithme de Hermes.

Nous avons, aussi, parlé d'autres techniques qui se passent de la résolution de l'équation aux dérivées partielles (EDP) et présentent, d'une nouvelle manière, la fonction distance [15]. En effet, les étudiants de notre département, ont utilisé deux méthodes de réduction de temps de calcul, pour différents domaines d'application.

Dans [22] le modèle adopté est basé sur le modèle de contour actif géodésique (CAG) [8], qui est basé contour, et le modèle de Chan et Vese (C-V) [21], qui est basé région, à la fois. Le modèle est implémenté implicitement par ensembles de niveaux (*Level set*). Pour rappel l'équation d'évolution du CAG est donnée sous la forme suivante :

$$\frac{\partial \Phi}{\partial t} = g |\nabla I| \left( \operatorname{div} \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) + c \right) + \nabla g \cdot \nabla I \quad (II.1)$$

où :  $g$  est, cette fois-ci, une fonction d'arrêt. Par contre, celle de C-V est donnée comme suit :

$$\frac{\partial \Phi}{\partial t} = d(\Phi) \left[ m \nabla \left( \frac{\nabla \Phi}{|\nabla \Phi|} \right) - v - I_1 (I - U)^2 + I_0 (I - V)^2 \right] \quad (II.2)$$

Avec :  $n, m, I_0, I_1$  des constantes positives

Le principe est de remplacer la fonction  $g$ , de l'équation (II.1), par une autre fonction dite la **force signée de pression (SPF)**. Cette dernière est une force générée par la différence de la moyenne des intensités à l'intérieur et à l'extérieur de l'objet ( $U$  et  $V$  respectivement) et leurs moyennes. Cette force applique une pression incitant le contour à ce rétrécir ou s'accroître selon son signe. La formule de la SPF est donnée par l'équation suivante :

$$SPF(I(v)) = \frac{I(v) - \frac{U+V}{2}}{\max \left( \left| I(v) - \frac{U+V}{2} \right| \right)}, \quad v \in \Omega \quad (II.3)$$

Après quelques restrictions apportées à la fonctionnelle [22], ils ont abouti au modèle suivant :

$$\frac{\partial \Gamma}{\partial t} = SPF(I(v)) \cdot a |\nabla I|, \quad v \in \Omega \quad (II.4)$$

où  $a$  : une constante qui est rajoutée pour augmenter la vitesse de l'évolution. Ce modèle est implémenté par les ensembles de niveaux, tout en sachant que la fonction *level set*  $\Phi$  ne doit prendre que deux valeurs possibles :

$\dot{\Phi} = +1$  pour toutes les valeurs positives de  $\Phi$ .

$\dot{\Phi} = -1$  pour toutes les valeurs négatives de  $\Phi$ .

$\dot{\Phi} = 0$  sur le contour.

Tous ceci a conduit à une réduction considérable du temps de calcul et a donné des bons résultats pour le tracking vidéo en temps réel [22].

Dans [23] le modèle adopté combine la force signée de pression [26] et la méthode *level set* accélérée [27]. La fonction *level-set* est définie [23] comme une fonction ayant des signes opposés à l'intérieur et à l'extérieur du contour  $\Gamma$  (figure II.1), elle que :

$$\Phi = \begin{cases} 3 & \text{pour les valeurs à l'extérieur de } \Gamma \\ 1 & \text{pour les valeurs appartenant à } L_{out} \\ -1 & \text{pour les valeurs appartenant à } L_{in} \\ -3 & \text{pour les valeurs à l'intérieur de } \Gamma \end{cases} \quad (II.5)$$

Avec :  $L_{in}$  un ensemble contenant les points sur les bords intérieur de  $\Gamma$ .

$L_{out}$  un ensemble contenant les points sur les bords extérieur de  $\Gamma$ .

Cette nouvelle présentation de la fonction *level-set* permet d'accélérer la convergence de l'évolution des contours actifs.

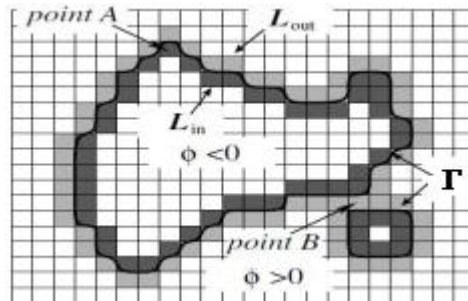


Fig .II.1. Représentation de la nouvelle fonction level set.

Par rappel, l'équation d'évolution de  $\Phi$  est donnée par l'équation II.1, où  $div(\frac{\nabla\Phi}{|\nabla\Phi|})$  est la courbure qui permet de garder une courbe lisse. Elle représente la  $F_{int}$ . Le reste de l'équation est la  $F_{ext}$ . Lorsque la vitesse  $F$  (qui représente la somme des  $F_{ext}$  et  $F_{int}$  (équation (II.2)) est négative, la valeur  $\Phi$  diminue localement, ce qui a pour effet de dilater le contour au point considéré. A l'inverse, le contour se rétracte si  $F > 0$ . Dans cette méthode seul le signe de  $F$  est pris en compte ( $\Phi$  réduite à un nombre restreint de valeurs, seuls les points voisins de l'interface sont concernés par l'évolution). A chaque itération  $F$  n'est calculé que pour les points  $L_{in}$  et  $L_{out}$ . Ainsi, les points de  $L_{out}$  passent dans  $L_{in}$  si  $F < 0$ , tandis que les points de  $L_{in}$  passent dans  $L_{out}$  si  $F \neq 0$ . Leur valeur est affecté à -1 ou 1 selon le cas. La région est ensuite mise à jour, de façon à ce qu'un point de  $L_{in}$  soit enlevé s'il n'a aucun voisin dans  $L_{out}$ , et inversement. De cette façon, les pixels changent rapidement de statut, et la courbe évolue de manière accélérée. L'application de ce modèle sur la segmentation 3D [23] a donné des résultats encourageants en ce qui concerne le résultat de segmentation et surtout en ce qui concerne la réduction du temps de calcul.

## II.3 Approche adoptée

### II.3.1 La fonction distance signée discrète

La fonction *level set* ( $\Phi$ ) est initialisée de telle sorte qu'elle représente la distance signée à un contour  $\Gamma$  (déjà initialisé). Elle est définie comme suit :

$$\Phi(x, y, t = 0) = \pm d \tag{II.6}$$

où  $d$  est la distance entre un point de l'image  $(x, y)$  par rapport à la courbe  $\Gamma$  (figure II.2 ), d'ou le nom : le fonction distance signée. Dans ce cas,  $\Phi$  est composée des réelles.

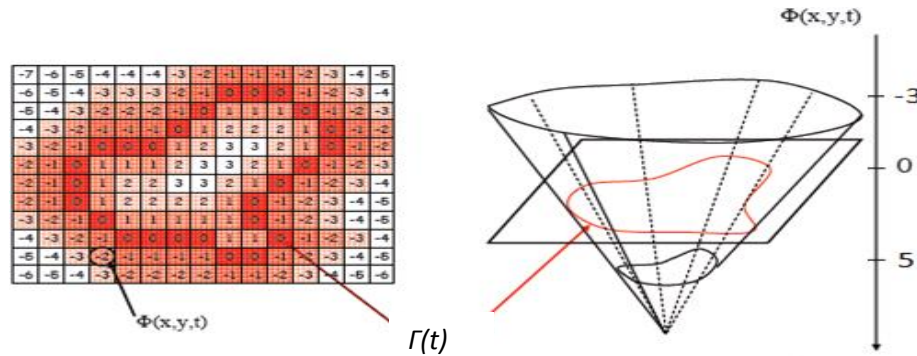


Fig. II.2. La représentation de la distance signée.

En changeant la représentation en réelles (figure II.3.a), pour la fonction distance signée, en représentation discrète, en utilisons les valeurs entières  $\{-1, 0, 1\}$  (figure II.3.b), on enlève une grande partie de la charge de calcul, tout en maintenant la précision de la solution.

Ainsi, le  $(-1)$  représente la région interne, le  $(0)$  représente le contour, et  $(1)$  représente la région externe [15].

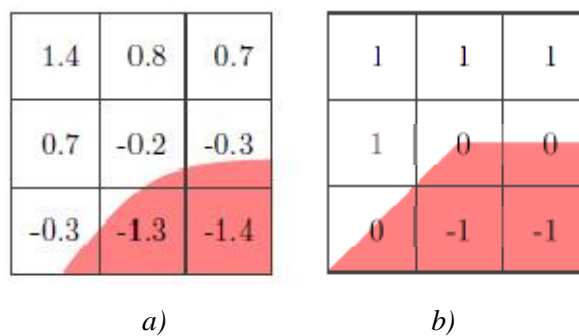
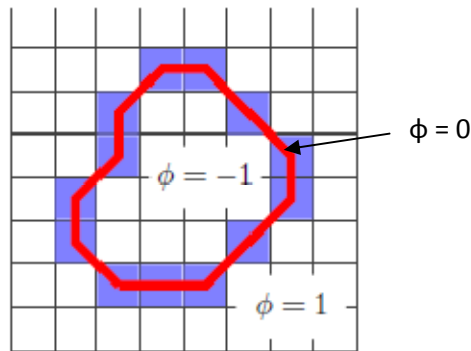


Fig. II.3. Une représentation d'une surface implicite sur une grille de 3x3: a) en valeurs réelles et b) discrète.

Cette méthode de représentation a plusieurs propriétés intéressantes. Tout d'abord, les calculs ne sont effectués que dans le niveau zéro, au lieu sur l'ensemble du domaine ou dans une petite bande étroite. Deuxièmement, la représentation approximative de la fonction de distance signée travaillera avec trois entiers (figure II.4).



**Fig. II.4.** Fonction distance signée discrète.

Troisièmement, la régularisation ne concerne que quelques calculs d'entiers et peut être manipulée en dehors de l'énergie elle-même, et donc éviter la résolution de la fonctionnelle. Quatrièmement, le niveau zéro est représenté exactement ce qui élimine la nécessité d'interpolation au cours de l'évolution de la courbe (revoir la figure II.3)

### II.3.2 Technique de l'évolution

L'algorithme principal (Algo. II.1) est constitué de deux phases : rétrécissement et dilatation. Ces deux étapes sont effectués sur un voisinage de 4 afin d'avoir une courbe minimale [15].

```

Initialisation de la fonction level set phi
Pour chaque itération faire
% Rétrécissement
  Calculer la force externe
  Prendre les valeurs positives de la force externe
  Propagation(Algo.2)
  Maintenir la courbe minimale (Algo.3)
% Dilatation
  Calculer la force externe
  Prendre les valeurs négatives de la force externe
  Propagation(Algo.2)
  Maintenir la courbe minimale (Algo.3)
FinPour

```

**Algo II.1.** Algorithme principal du modèle adopté.

Pour chaque itération de l'évolution, le programme rappelle le calcul de la force externe pour tous les points de la courbe, permettant ainsi de savoir le sens d'évolution (rétrécissement ou dilatation).

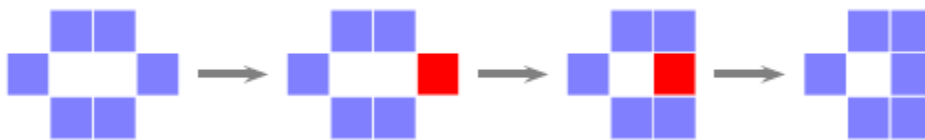
La force externe peut-être calculer et utiliser par des méthodes différentes (section II.3). La courbe évolue (Algo. II.2) selon le signe de la force externe sur tous les points de la courbe.

```

Pour chaque point x sur la courbe faire
  Si la force externe est non nulle alors
    Si la force externe est négative alors
       $\Phi(x) \leftarrow 1$ 
    Sinon  $\Phi(x) \leftarrow -1$ 
  Fin si
  Supprimer x de la courbe
  Pour chaque voisin y de x ayant un signe opposé à  $\Phi(x)$  faire
     $\Phi(y) \leftarrow 0$ 
    Insérer y à la courbe
  Fin pour
Fin si
Fin pour
  
```

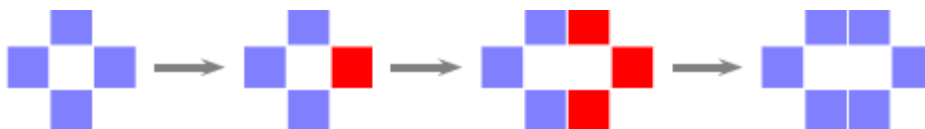
**Algo II.2.** La propagation de la courbe.

les points en question seront supprimés de la courbe et seront considérés de l'extérieur pour une force négative et de l'intérieur pour une force positive. Ainsi ces points recevront la valeur 1 ou -1, respectivement, et la valeur zéro sera attribuée à un point voisin ayant signe opposé (figure II.5 et figure II.6).



**Fig. II.5.** L'étape de rétrécissement.

Dans la figure II.5, le pixel rouge a une force positive. Il a tendance de se déplacer vers l'intérieur et se positionner sur un pixel voisin ayant un signe opposé (Algo. II.2).



**Fig. II.6.** L'étape de dilatation.

Dans la figure 6, le pixel rouge a une force négative. Il a tendance de se déplacer vers l'extérieur et se positionner sur un pixel voisin ayant un signe opposé (Algo. II.2).

Notant que, après chaque phase, il est important d'éliminer les points non nécessaires pour maintenir une courbe minimale (Algo. II.3) afin d'éviter les artéfacts [1]. Avec une courbe minimale, on a besoin d'utiliser un voisinage de 4 voisins (figure II.7).



**Fig. II.7.** Elimination des points durant l'évolution.

```

Pour chaque  $x$  de la courbe faire
  Si tout les voisins  $y$  de  $x$  tel que  $\phi(y) \geq 0$ , alors
     $\phi(x) \leftarrow 1$ 
    Supprimer  $x$  de la courbe
  Fin si
  Si tout les voisins  $y$  de  $x$  tel que  $\phi(y) \leq 0$ , alors
     $\phi(x) \leftarrow -1$ 
    Supprimer  $x$  de la courbe
  Fin si
Fin pour
  
```

**Algo II.3.** Maintenir la courbe minimale.

### II.3.3 Critère d'arrêt

Pour le critère d'arrêt, on peut fixer le nombre d'itérations. On peut, aussi, vérifier si la fonction *level set* ne change pas de valeurs ou il y a un changement très minime de valeurs, ce qui est notre cas.

Mais, comme on pouvait le constater, l'évolution de la courbe ressemble à celle des contours actifs paramétriques, on peut, donc, examiner le nombre de points de la courbe qui changent de valeurs et de fixer un seuil pour ce nombre

## II.4 les différentes forces externes

Il existe plusieurs types de forces externes, qui représentent l'information image dans l'approche adoptée. On citera, entre autres, celle qui consiste à minimiser la distance de bahattacharyya entre deux distributions d'intensité [15], et celle qui propose une fonctionnelle d'énergie qui combine deux techniques de segmentation, appliquée séparément : le seuillage

d'intensité et le contour actif géométrique [22 ; 23]. Dans notre cas, nous intéresserons à la moyenne et la variance des régions.

### II. 4.1 la moyenne d'intensité

La stratégie de segmentation consiste à séparer l'intensité moyenne de deux régions. L'énergie peut-être représentée par [21]:

$$E(x) = \int (I(x) - V)^2 H(f(x)) + (I(x) - U)^2 H(f(x)) dx \quad (II.6)$$

H(.) étant la fonction Heaviside. Les moyennes d'intensité des pixels situés à l'intérieur et l'extérieur de la courbe  $\Gamma$  [15], notée par U et V respectivement, peuvent être calculées à chaque itération de l'évolution comme :

$$U = \frac{\int_{\Omega} I(x) H(f(x)) dx}{\int_{\Omega} H(f(x)) dx} \quad V = \frac{\int_{\Omega} I(x)(1 - H(f(x))) dx}{\int_{\Omega} (1 - H(f(x))) dx} \quad (II.7)$$

Ces statistiques portent des informations précieuses sur les statistiques de l'objet et le fond.

La forme simple du gradient de l'équation (II.6) est :

$$\nabla E(x) = (I(x) - V)^2 - (I(x) - U)^2 \quad (II.8)$$

Dans nos tests, nous avons pris cette forme pour sa simplicité.

### II.4.2 La moyenne et la variance

La séparation par la moyenne de l'intensité assure seulement la segmentation pour les distributions de variance unité ( $\delta = 1$ ), une méthode qui conduit souvent à une segmentation incorrecte. Pour cette raison, dans [15] ils introduisent une autre énergie :

$$E(x) = \int \log \mathfrak{N}(U, d_u) H(f(x)) + \log \mathfrak{N}(V, d_v) H(f(x)) dx \quad (II.9)$$

ou:  $\mathfrak{N}(U, \delta_u)$  et  $\mathfrak{N}(V, \delta_v)$  sont des distributions normales [15] et  $\delta_u$  et  $\delta_v$ , les variances intérieure et extérieure, respectivement:

$$d^2_u = \frac{\int_{\Omega} (I(x) - U)^2 H(f(x)) dx}{\int_{\Omega} H(f(x)) dx} \quad d^2_v = \frac{\int_{\Omega} (I(x) - V)^2 H(1 - f(x)) dx}{\int_{\Omega} H(1 - f(x)) dx} \quad (II.10)$$

Cette énergie est une intégration du log-likelihood (log de la vraisemblance) des distributions  $\mathfrak{N}$  qui permet une segmentation par la moyenne et la variance. Son gradient peut s'écrire comme suite :

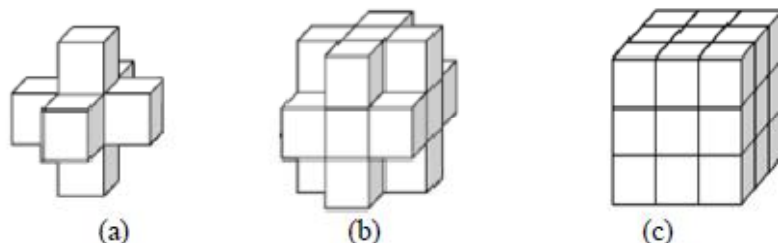
$$\nabla E(x) = \left( \frac{I(x) - V}{d_v} \right)^2 - \left( \frac{I(x) - U}{d_u} \right)^2 + \log \left( \frac{d_v}{d_u} \right) \quad (II.11)$$

Si  $\delta v = \delta u = 1$  on aura:

$$\nabla E(x) = (I(x) - V)^2 - (I(x) - U)^2 \quad (II.12)$$

## II.5 Extension aux trois dimensions

Dans les applications tridimensionnelles, la scène est représentée par un tableau à 3 dimensions de volumes élémentaires appelé *voxel*, généralement de forme cubique. Un *voxel* est composé : de surfaces, d'arêtes et de sommets [23]. Cela permet de définir un voisinage *n-connexe* ( $n = 6, 18$  ou  $26$ ) du *voxel P* (figure II.8).



**Fig. II.8.** Voisinage d'un voxel : (a) 6-voisins , (b)18-voisins et (c) 26-voisins.

Dans notre cas, et selon l'exigence d'une courbe minimale, on va inclure deux points supplémentaires pour tenir compte de la dimension ajoutée. En d'autres termes, on va utiliser un voisinage de 6 afin d'avoir une surface minimale, comme dans le 2D ou on a utilisé un voisinage minimum de 4.

## II.6 complexité algorithmique du modèle

La complexité algorithmique permet de mesurer les performances d'un algorithme et de le comparer avec d'autres algorithmes réalisant les mêmes fonctionnalités. Elle est un concept fondamental pour tout informaticien, elle permet de déterminer si un algorithme (a) est meilleur qu'un algorithme (b) et s'il est *optimal*. En effet, différents algorithmes ont des coûts différents en termes de temps d'exécution (nombre d'opérations effectuées par

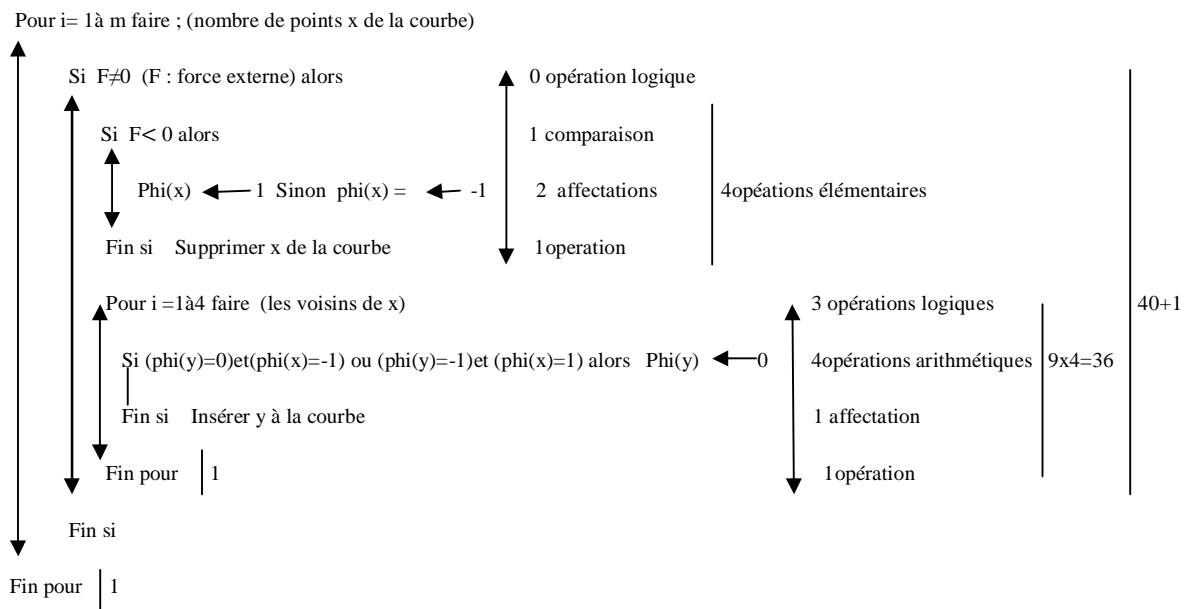
l'algorithme) et de taille mémoire (taille nécessaire pour stocker les différentes structures de données pour l'exécution). Ces deux concepts sont appelés la complexité en temps et en espace de l'algorithme. Dans notre cas nous nous intéressons à la complexité en temps.

Afin de pouvoir calculer cette complexité, plusieurs règles sont envisageables et fondamentaux :

1. le temps d'exécution (t.e.) d'une affectation ou d'un test est considéré comme constant.
2. le temps d'une séquence d'instructions est la somme des instructions qui la composent,
3. le temps d'un branchement conditionnel est égal au t.e. du test plus le max des deux t.e. correspondant aux deux alternatives (dans le cas d'un temps max).
4. Le temps d'une boucle est égal à la somme du coût du test, du corps de la boucle et du test de sortie de boucle.

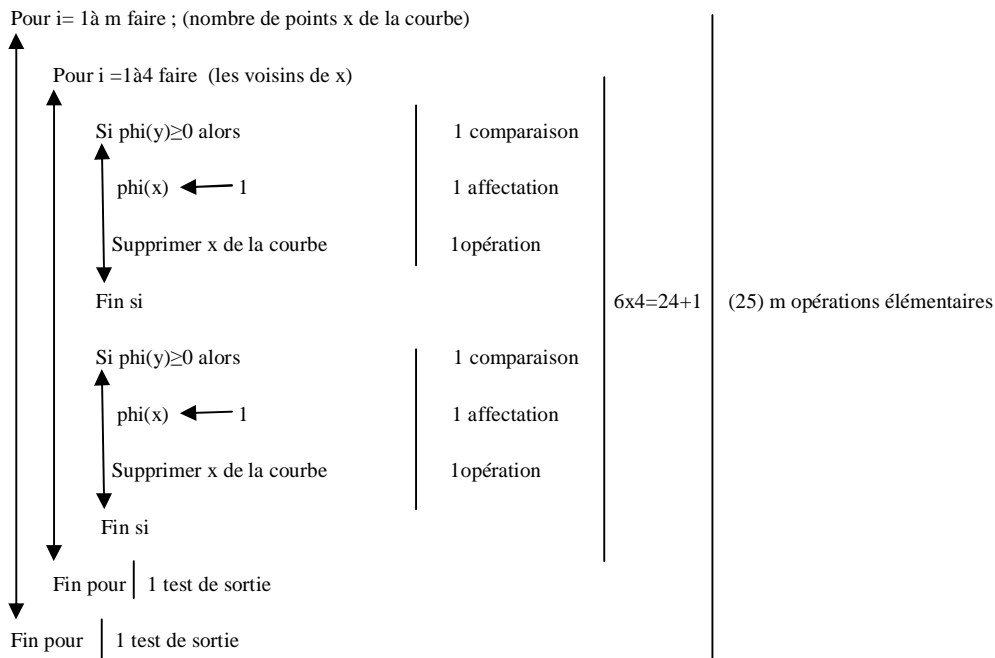
La notation de la complexité est donnée par :  $O(\cdot)$ . Les principales complexités sont :  $O(1)$  : temps constant,  $O(\log(n))$  : complexité logarithmique,  $O(n \log(n))$ ,  $O(n)$  : complexité linéaire,  $O(n^2)$   $O(n^3)$  ...  $O(n^p)$  polynomiales et  $O(p^n)$  complexité exponentielle, pour  $n$  itérations.

**Algo II.2**



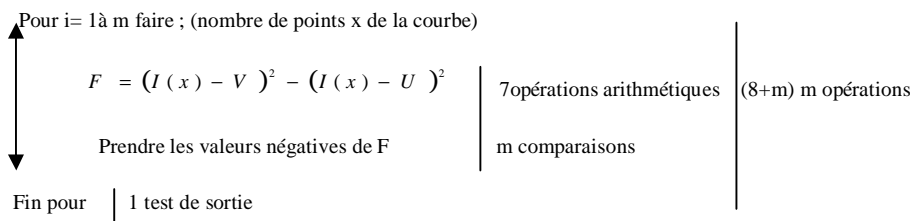
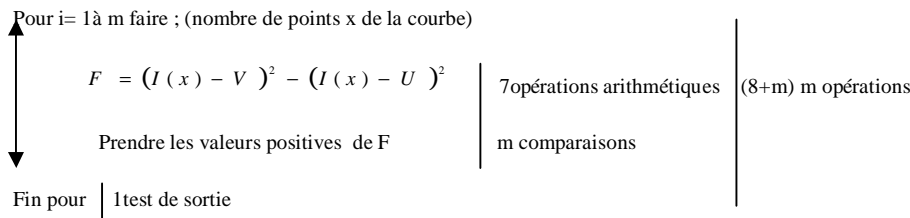
La complexité algorithmique de **Algo II.2** est de 41 opérations élémentaires multiplier par m nombre de points sur la courbe, donc  $(41m+1$  opération de sortie) opérations élémentaires.

**Algo II.3**



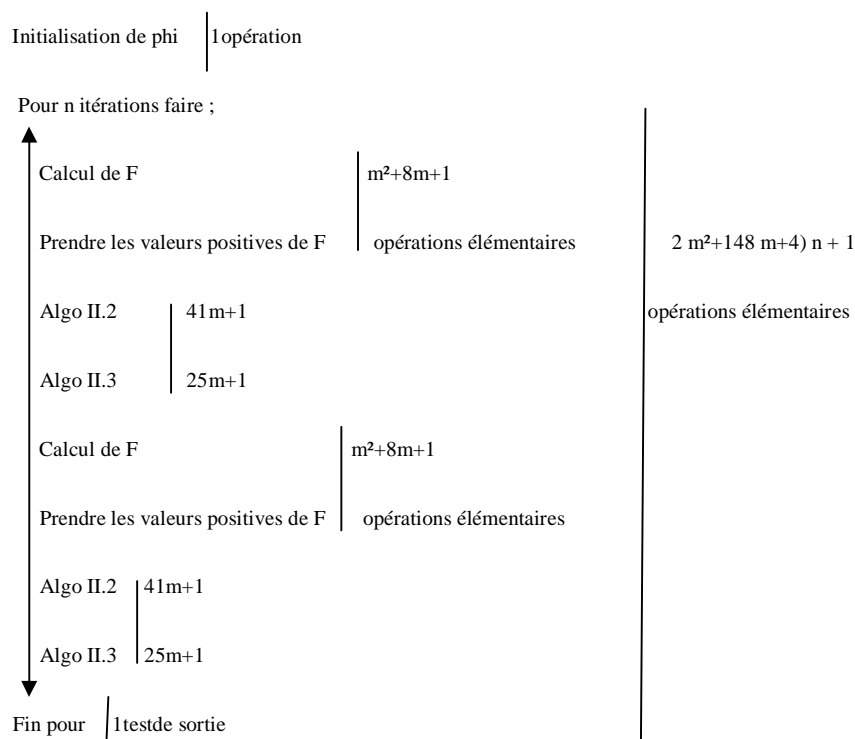
La complexité algorithmique de **Algo II.3** est de 25 opérations élémentaires multiplier par m nombre de points sur la courbe, donc (25m+1 opération de sortie) opérations élémentaires.

**Calcul de la force externe F**



La complexité algorithmique de calcul de la force externe est de (m+8) opérations élémentaires multiplier par m nombre de points sur la courbe, donc ((m<sup>2</sup>+8m)+1 opération de sortie) opérations. Puisque nous calculons la force externe deux fois (pour l'étape de dilatation et l'étape de rétrécissement) alors sa complexité algorithmique devient 2(m<sup>2</sup>+8m+1).

**Algo II.1**



La complexité finale est  $(2 m^2 + 148 m + 4)n + 1$ . Le terme  $C=2 m^2 + 148 m + 4$  est dépendant de la taille de l'image et du contour initial, non pas du nombre d'itérations, donc la complexité algorithmique peut s'écrire sous la forme  $Cn+I$ . Sachant qu'il existe  $C'$  tel que  $C'n > Cn+I$  donc, la complexité algorithmique de l'approche adoptée est linéaire  $O(n)$ .

**II.7 Conclusion**

Dans ce chapitre, nous avons décrit, d'une manière assez précise, les différentes étapes du modèle de contour actif accéléré que nous avons adopté.

Nous pouvons bien constater que c'est un modèle très simple et qui réduit, considérablement, le temps de convergence. Cependant, nous pourrions conclure que c'est un modèle qui évolue localement. Pour cela, nous aurons, probablement, des problèmes d'initialisations et d'évolutions sur des images bruitées. Tout ceci fera l'objet des tests dans le chapitre suivant.

### **III.1 Introduction**

Dans ce chapitre, nous en proposons de tester l'approche adoptée sur différentes images. Les tests concernent le choix de la forme du contour initial. Plusieurs formes de contours initiaux seront testées pour avoir une idée de l'initialisation optimale, ainsi accélérer davantage l'approche. Une fois que le choix est fait nous le testons sur d'autres images (niveau de gris, couleur, bruitées et 3D), afin d'assurer que c'est le bon choix. Ensuite, un test comparatif entre l'approche adoptée avec d'autres approches sera effectué. Le test concernera la qualité de la segmentation et la vitesse de l'évolution.

### **III.2 Matériel utilisé et présentation des images**

#### **III.2.1 Matériel utilisé**

Notre matériel se résume à un Pc portable HP: AMD Athlon™ Néo processor MV-40 1.60GHz avec une mémoire (RAM): 2.00 Go (1.87 Go utilisable).

Notre application est développée sous Matlab : version 7.

#### **III.2.2 Présentation des images utilisées**

Dans cette partie nous présentons les images que nous allons utiliser pour nos différents tests. La diversité de cette collection et les tests effectués nous permettront d'avoir une idée large et précise sur les performances de l'approche adoptée.

Les images sont regroupées en images niveaux de gris (figure III.1), images couleurs (figure III.2), et images bruitées (figure III.3). Pour ce qui concerne les images 3D, elles seront présentées dans la partie III.3.

Les images en niveaux de gris sont en nombre de 8, et ont pour taille(en pixel) : Image1 :400X320, Image2 :300X246, Image3 :308X242, Image4 : 256X128, Image5: 256X256, Image6 : 256X256, Image7 : 512X384 et Image15 : 125X125.

Les images en couleur sont en nombre de 3 et de taille(en pixel) : Image8 : 381X321, Image9 : 320X256, et Images10 : 481X321.

Les images bruitées sont en nombre de 4 et de taille(en pixel) : Image11 : 300X200, Image12 : 256X256, Image13 : 98X90, et Image14 : 82X82. Pour ces dernières l'intensité du bruit diffère d'une image à une autre.

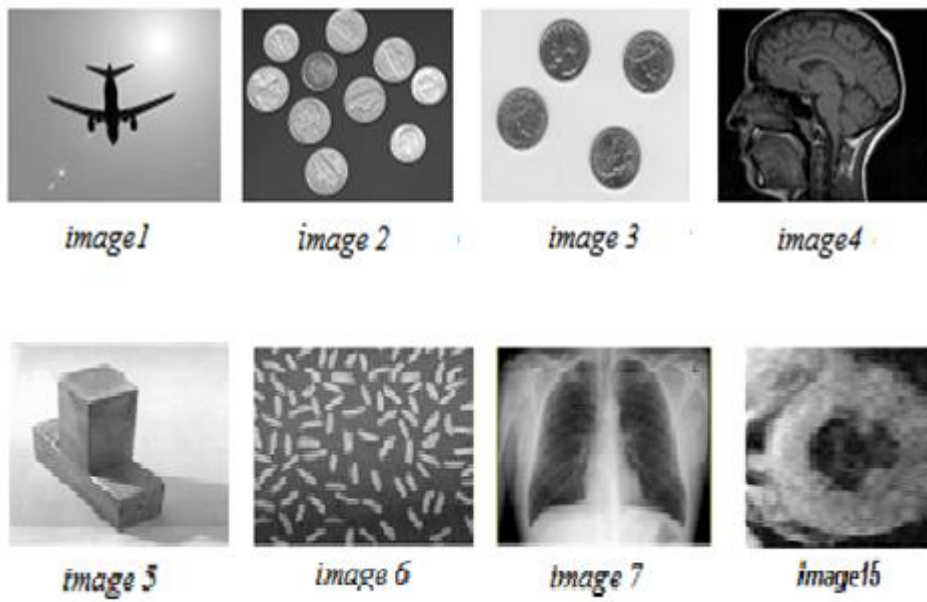


Fig.III.1. Images en niveaux de gris.



Fig.III.2. Images en couleur.

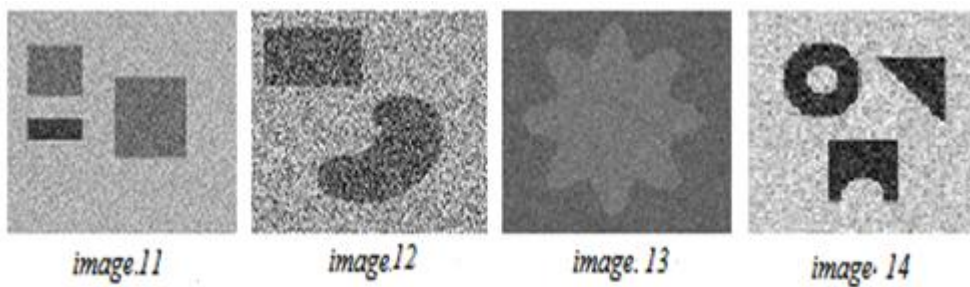
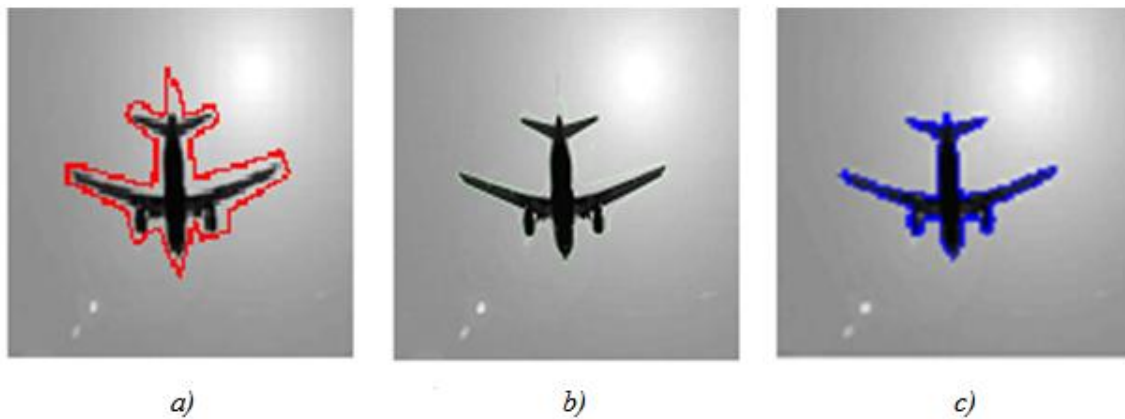


Fig.III.3. Images bruitées.

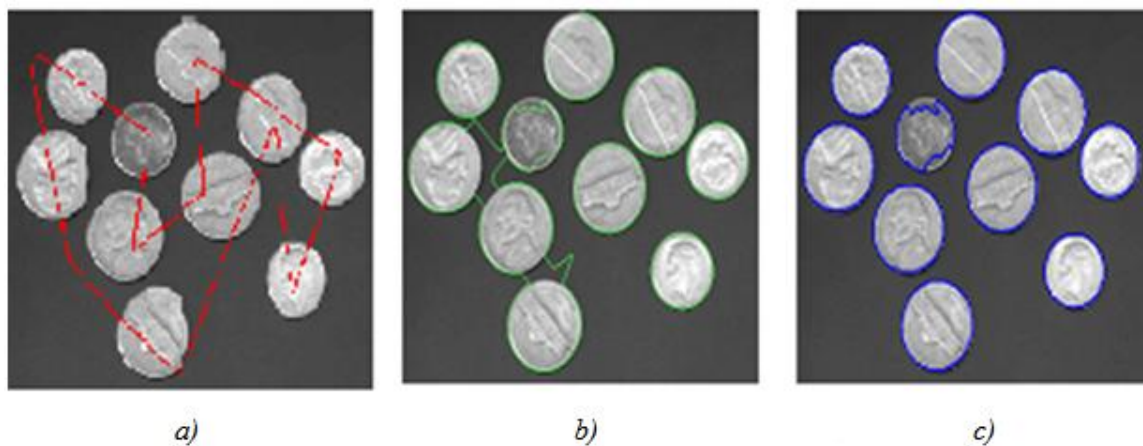
### III.3 Tests et résultats

#### III.3.1 Tester le contour initial

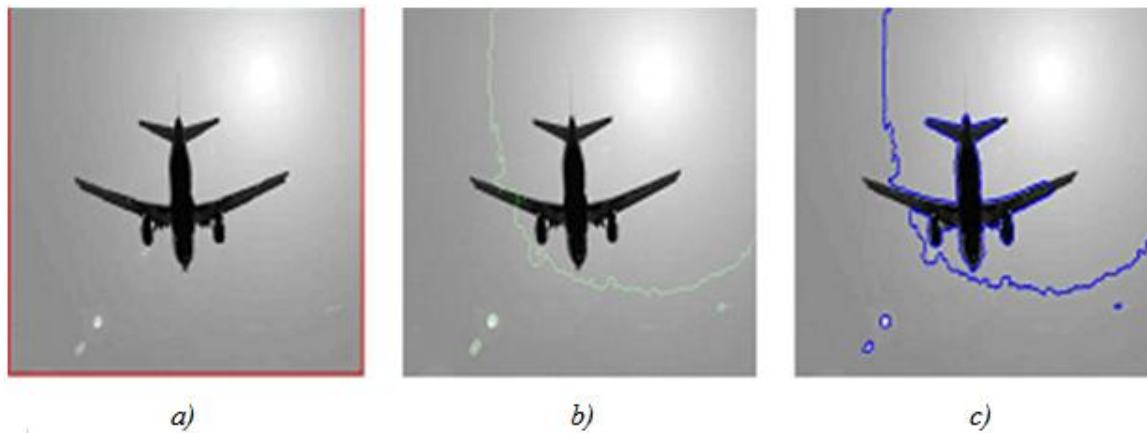
En appliquant cinq différentes initialisations (manuelle (figure III.4 et figure III.5), rectangle (figure III.4 et figure III.5), cercle (figure III.4 et figure III.5), 4 cercles (figure III.4 et figure III.5), 16 cercles (figure III.4 et figure III.5)) sur les images 1 et 2, nous déterminerons l'initialisation que nous utiliserons dans les prochains tests.



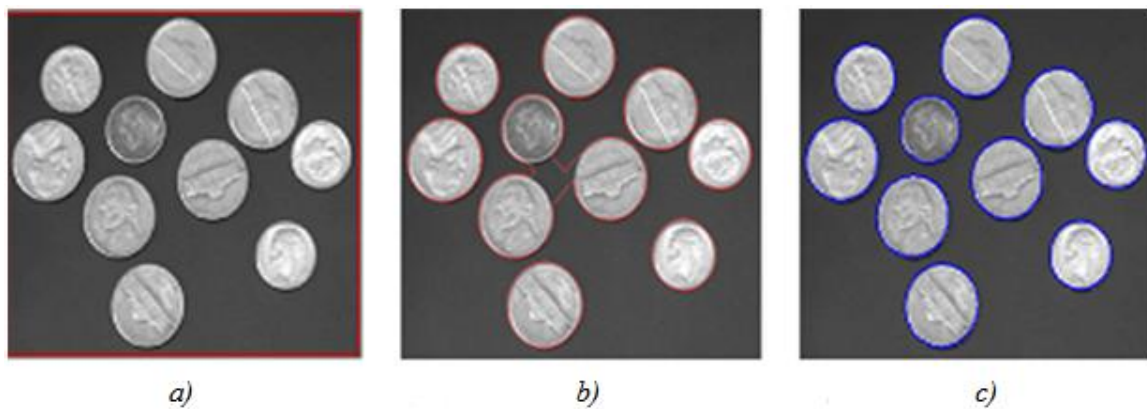
**Fig.III.4.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration131 et c)résultat final (temps de l'évolution 17.6233s).



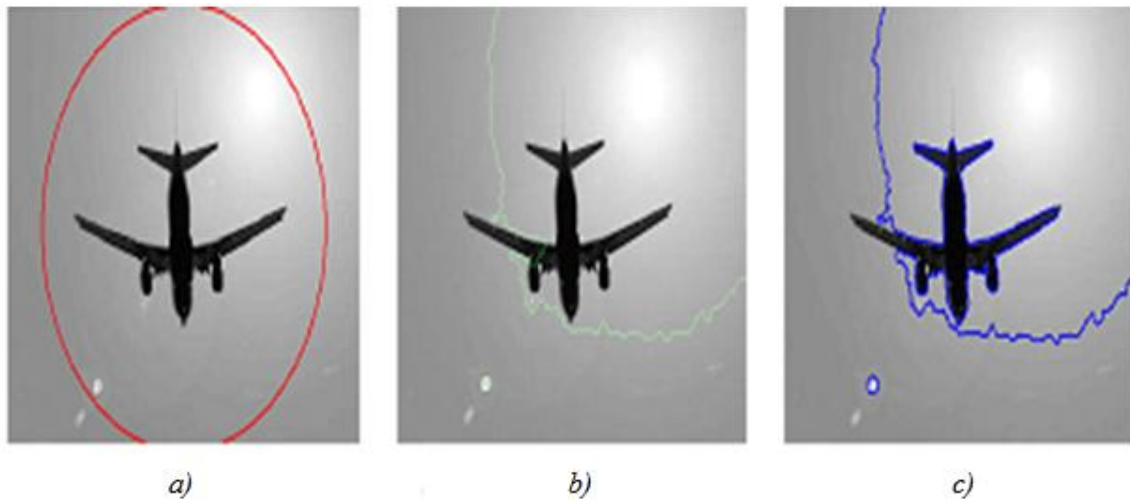
**Fig.III.5.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration13 et c)résultat final (temps de l'évolution 20.7358s).



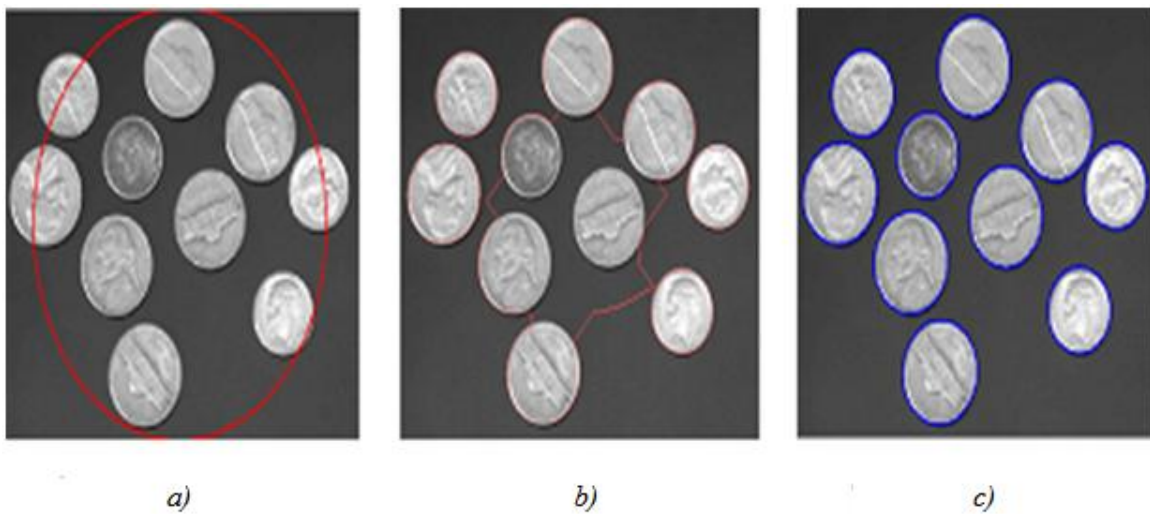
**Fig.III.6.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration131 et c)résultat final (temps de l'évolution 240.9656 s).



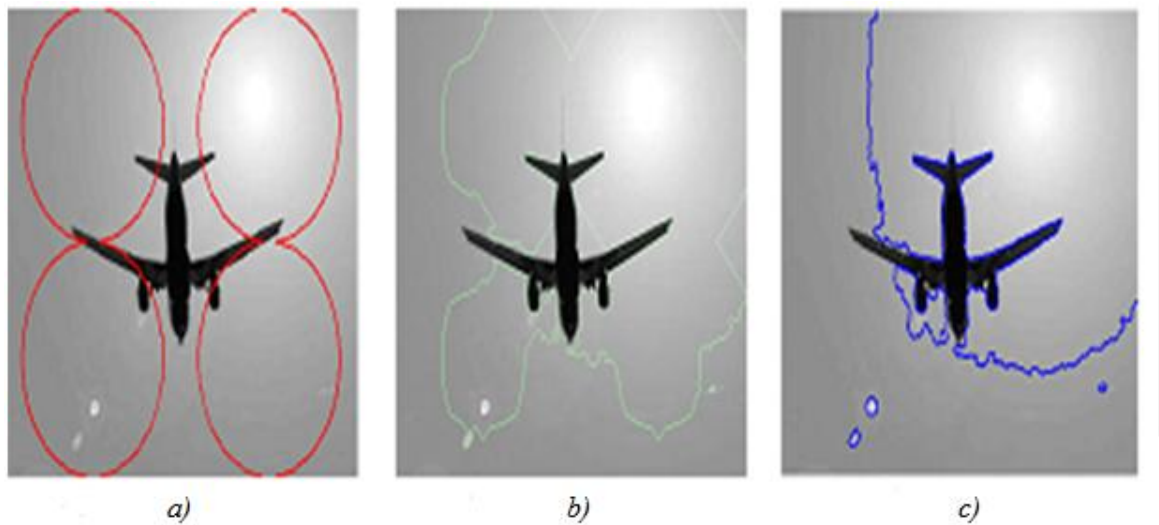
**Fig.III.7** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration131 et c)résultat final (temps de l'évolution 57.1195s).



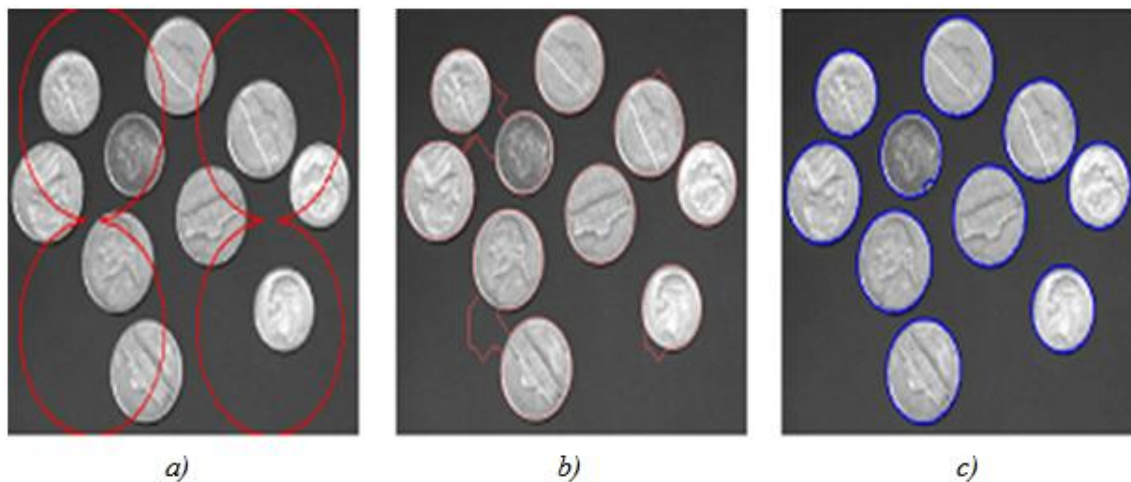
**Fig.III.8.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration134 et c)résultat final (temps de l'évolution 125.5857s).



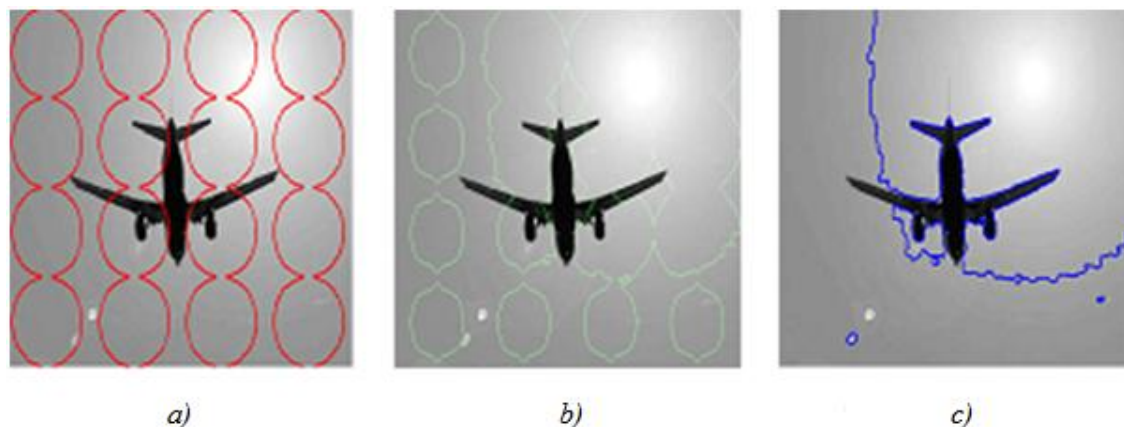
**Fig.III.9.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration71 et c)résultat final (temps de l'évolution 54.9873 s).



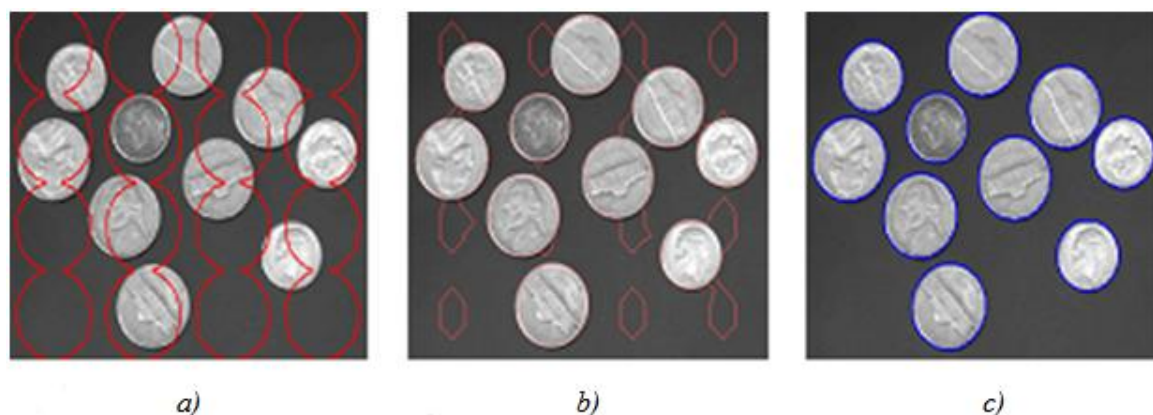
**Fig.III.10.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration33 et c)résultat final (temps de l'évolution est de 47.2346 s).



**Fig.III.11.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration49 et c)résultat final (temps de l'évolution 25.295s).



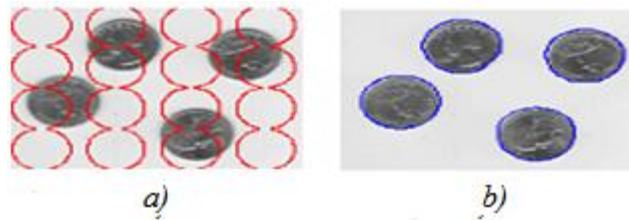
**Fig.III.12.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration10 et c)résultat final (temps de l'évolution 31.1065s).



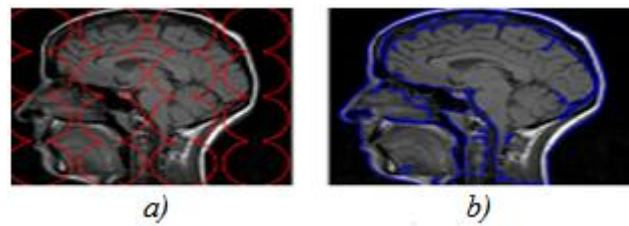
**Fig.III.13.** Résultat de segmentation : a)initialisation, b)segmentation a l'iteration20 et c)résultat final (temps de l'évolution 16.1003s).

Les résultats de ces tests nous conduisent à conclure que toutes les initialisations donnent un temps et une qualité de segmentation acceptable qui diffèrent d'une image à une autre et d'une initialisation à une autre. Cependant, nous remarquons qu'une initialisation manuelle très proche de l'objet a donné un résultat meilleur en temps et en qualité (figure III.4). Seulement, pour plusieurs objets, l'initialisation manuelle s'avère délicate (figure III.5).

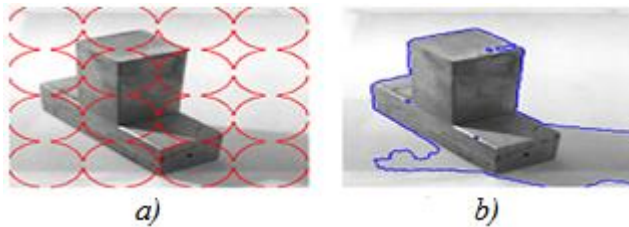
Toutefois, nous avons constaté dans les différents tests qu'il y a un léger avantage, dans la vitesse et la qualité, pour une initialisation à 16 cercles. Cela s'explique par le fait qu'on peut avoir une initialisation qui approche les contours des objets à segmenter. En conséquence, pour ne pas plonger dans le doute, nous avons testé cette initialisation sur d'autres images en niveaux de gris (figure III.14, figure III.15, figure III.16 et figure III.17), en couleur (figure III.18, figure III.19 et figure 20) et bruitées (figure III.21, figure III.22, figure III.23, et figure III.24).



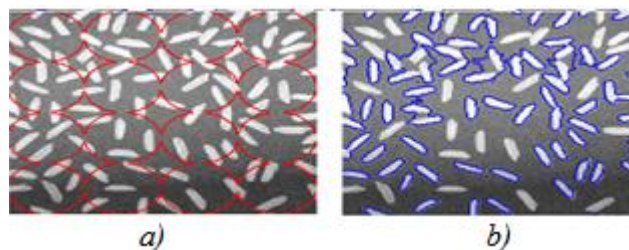
**Fig.III.14.** Résultat de segmentation : a)initialisation et b)résultat final (temps de l'évolution 12.8305s, itération 35).



**Fig.III.15.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 9.6851s, itération 64).



**Fig.III.16.** Résultat de segmentation : a)initialisation et b)résultat final (temps de l'évolution 10.9871s, itération 45).



**Fig.III.17.** Résultat de segmentation : a)initialisation et b)résultat final (temps de l'évolution 22.1207s, itération 50).

Les résultats sur les images en niveaux de gris s'avère satisfaisants dans les figures III.14, III.15 et III.16, mais moins satisfaisant dans la figure III.17. Ce résultat ne nous a pas surpris, nous savons que l'approche évolue localement et une initialisation près de l'objet

est recommandée. En effet, dans la figure III.17, les objets sont très petits, de ce fait il faut plus de cercles afin d'avoir une initialisation proche de tous les objets.



**Fig.III.18.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 242.6104s, itération 300).

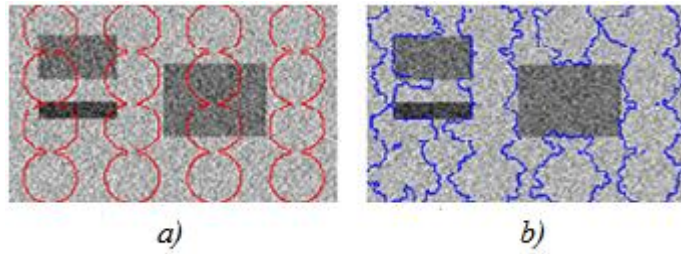


**Fig.III.19.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 82.7306s, itération 300).

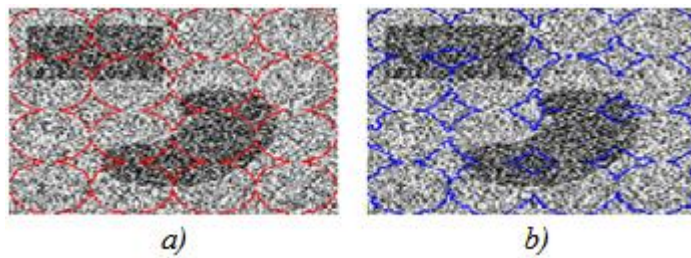


**Fig.III.20.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 315.5808s, itération 300).

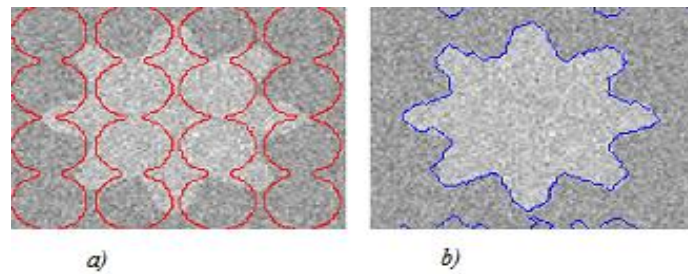
Les résultats obtenus sur les images couleurs sont très satisfaisants en termes de temps et qualité de la segmentation.



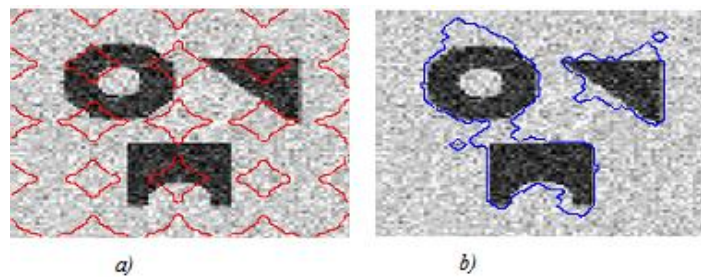
**Fig.III.21.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 8.1595s, itération 300).



**Fig.III.22.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 0.59441s, itération 27).



**Fig.III.23.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 0.1042s, itération 17).

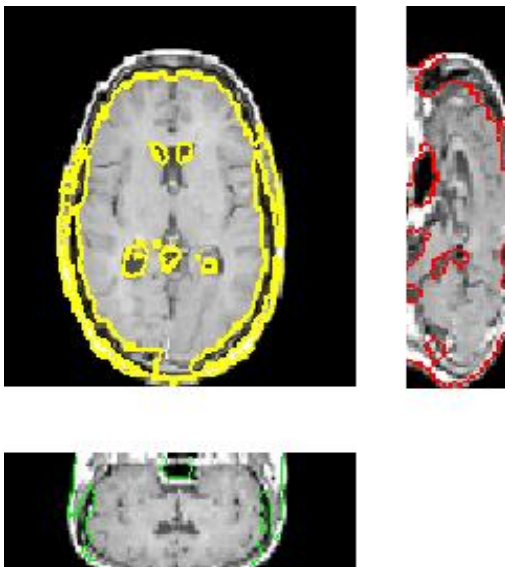


**Fig.III.24.** Résultat de segmentation : a)initialisation et b) résultat final (temps de l'évolution 0.080938 s, itération 23).

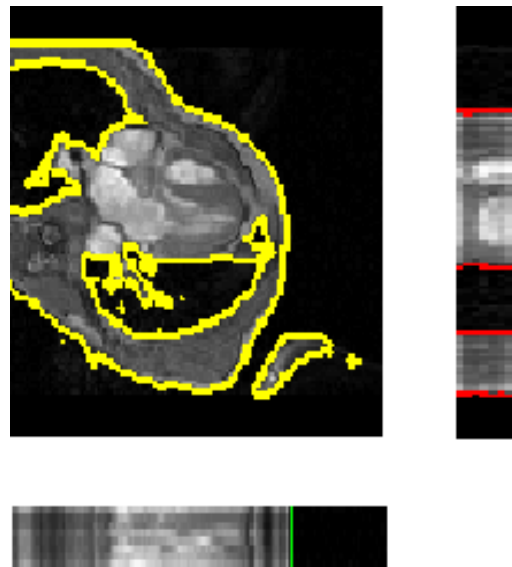
Les résultats les images bruitées sont moins bons. D'ailleurs, plus l'image est bruitée plus le résultat est de moins au moins bons. L'approche évolue localement et dans ce cas là la courbe s'accroche sur les minimums locaux. Nous pouvons suggérer qu'un ajout d'un filtre pour le lissage du bruit est très important.

### III.3.2.4 Application sur des images 3D

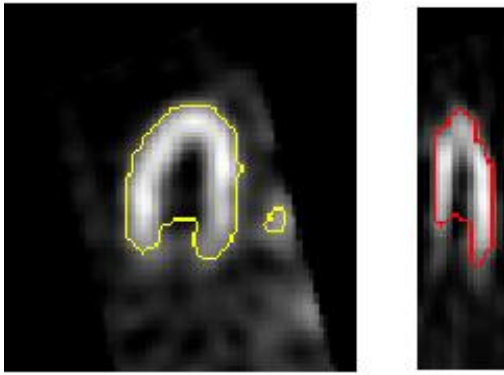
Nous avons utilisé 6 différentes images médicales 3D, prises sous forme de coupes : l'image IRM de Matlab de taille 128x128x27 (figure III.25 (a)), image IRM heart1 de taille 156x256x16 (figure III.25 (b)), image IRM heart2 de taille figure III.26 (a), image échographie de taille 120x128x8 (figure III.26 (b)), image IRM cérébrale de taille 512x512x25 (figure III.27 (a)) et image IRM thorax de taille 70x70x264 (figure III.27 (b)).



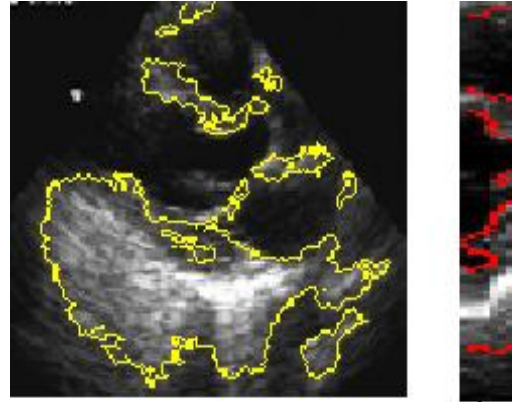
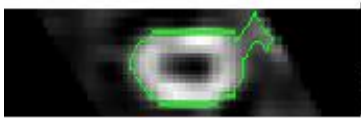
**Fig III.25.** Résultat après 300 itérations et un temps de 922.04s).



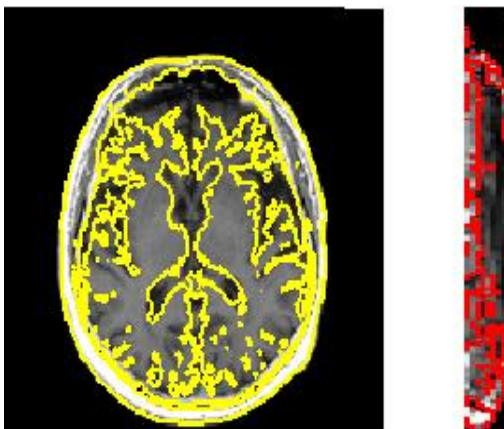
**Fig III.26.** Résultat après 300 itérations et un de temps 1186.118s).



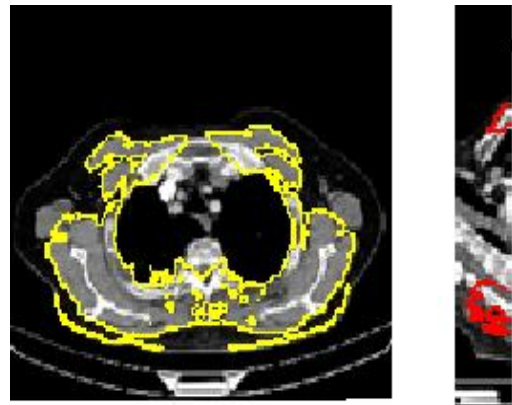
**Fig III.27.** Résultat après 300 itérations  
et un temps de 101.868s.



**Fig III.28.** Résultat après 300 itérations  
et un temps de 226.669s.



**Fig. III.29.** Résultat après 300 itérations  
et un temps de 7284.621 s.



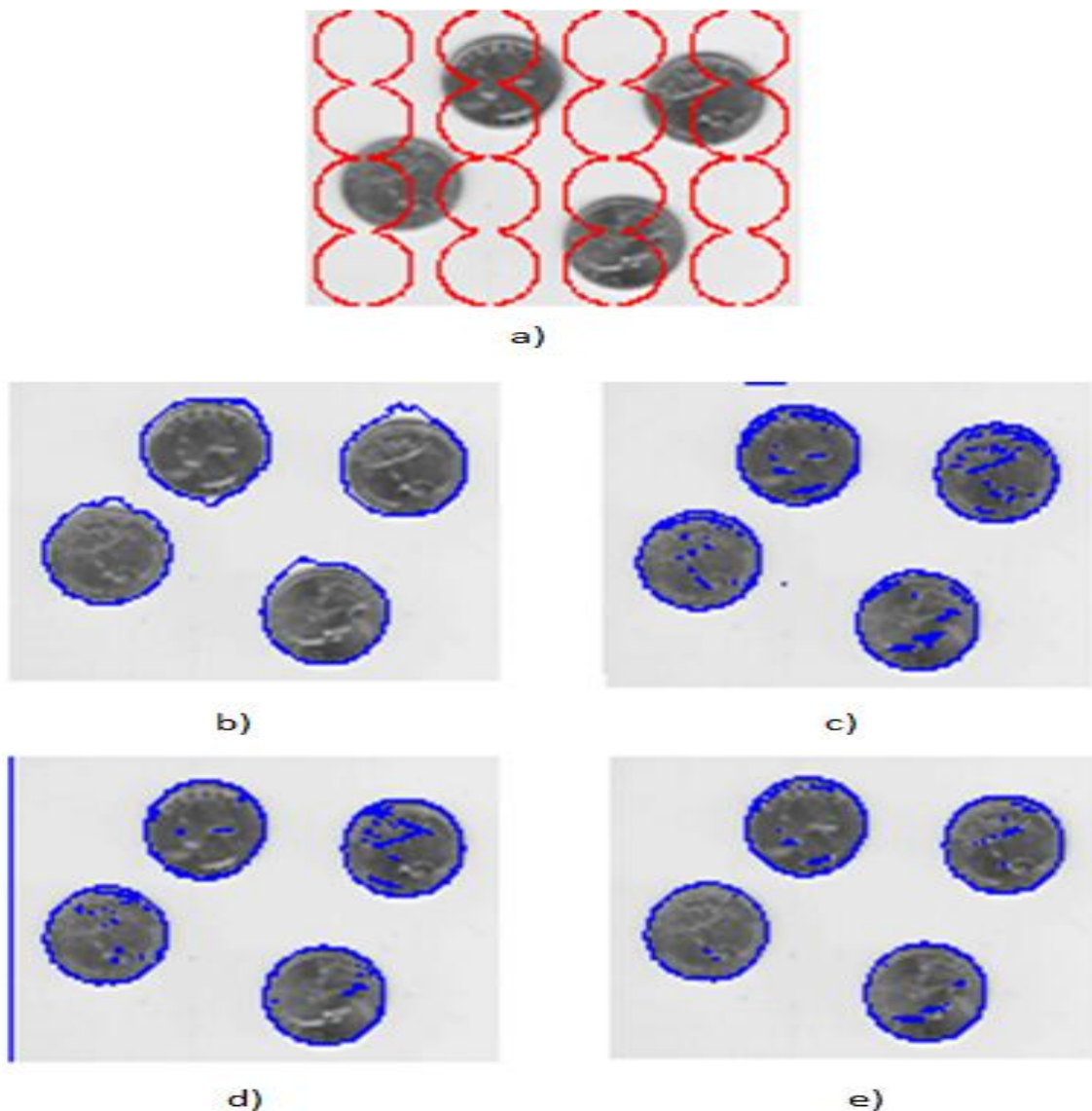
**Fig. III.30.** Résultat après 300 itérations  
et un temps de 8573.741 s.



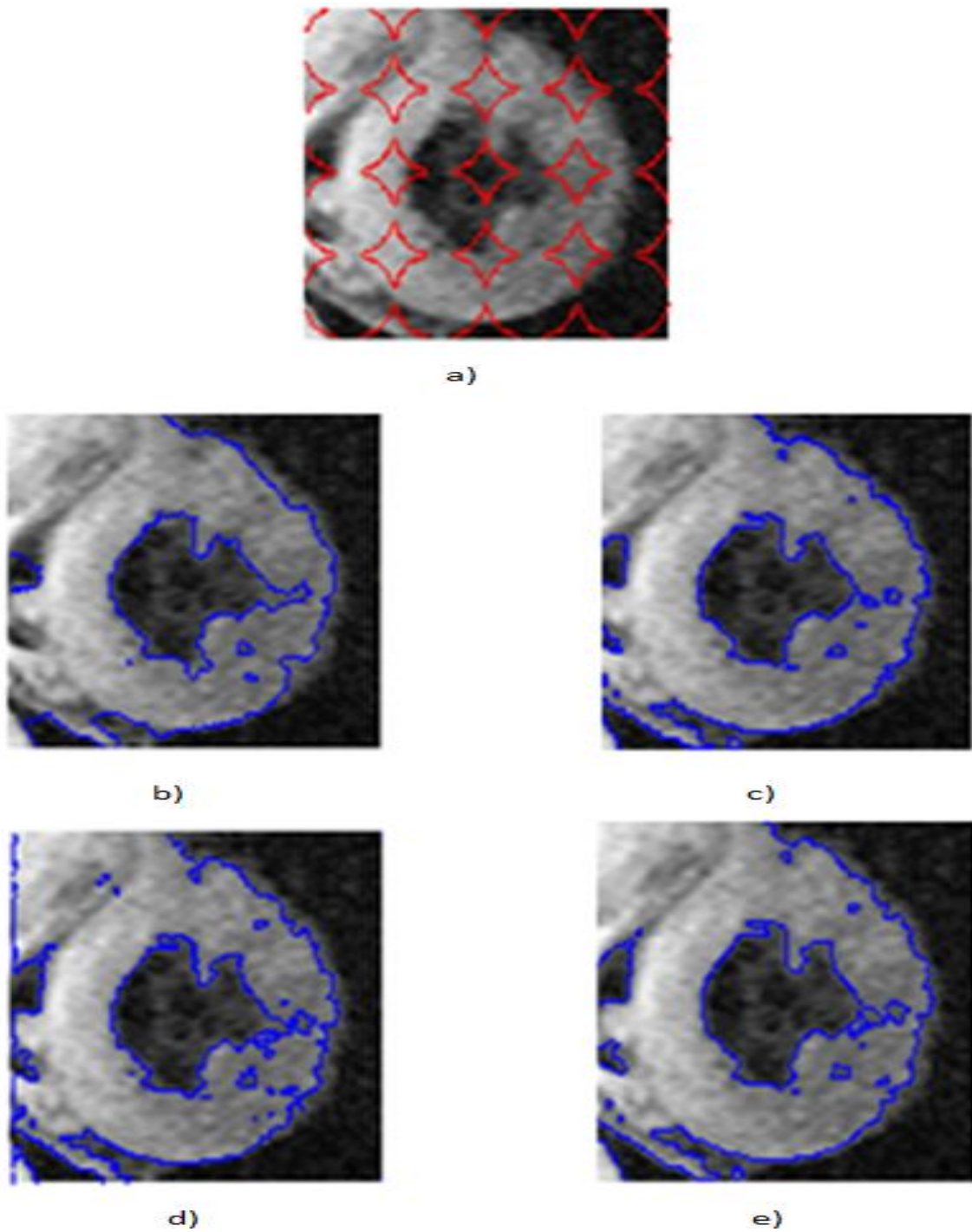
Les résultats obtenus sur les images 3D sont très bons en termes de temps et qualité de la segmentation par rapport aux résultats obtenus dans [23]. Par exemple, pour l'images IRM de Matlab le temps de convergence est moins de 10 minutes (922.04s) après 300 itérations par contre dans [23] le temps est un peu plus de 8 heures et 30 minutes (30816 s) après 100 itérations seulement, pour le même résultat de segmentation.

### III.4 Evaluation comparative

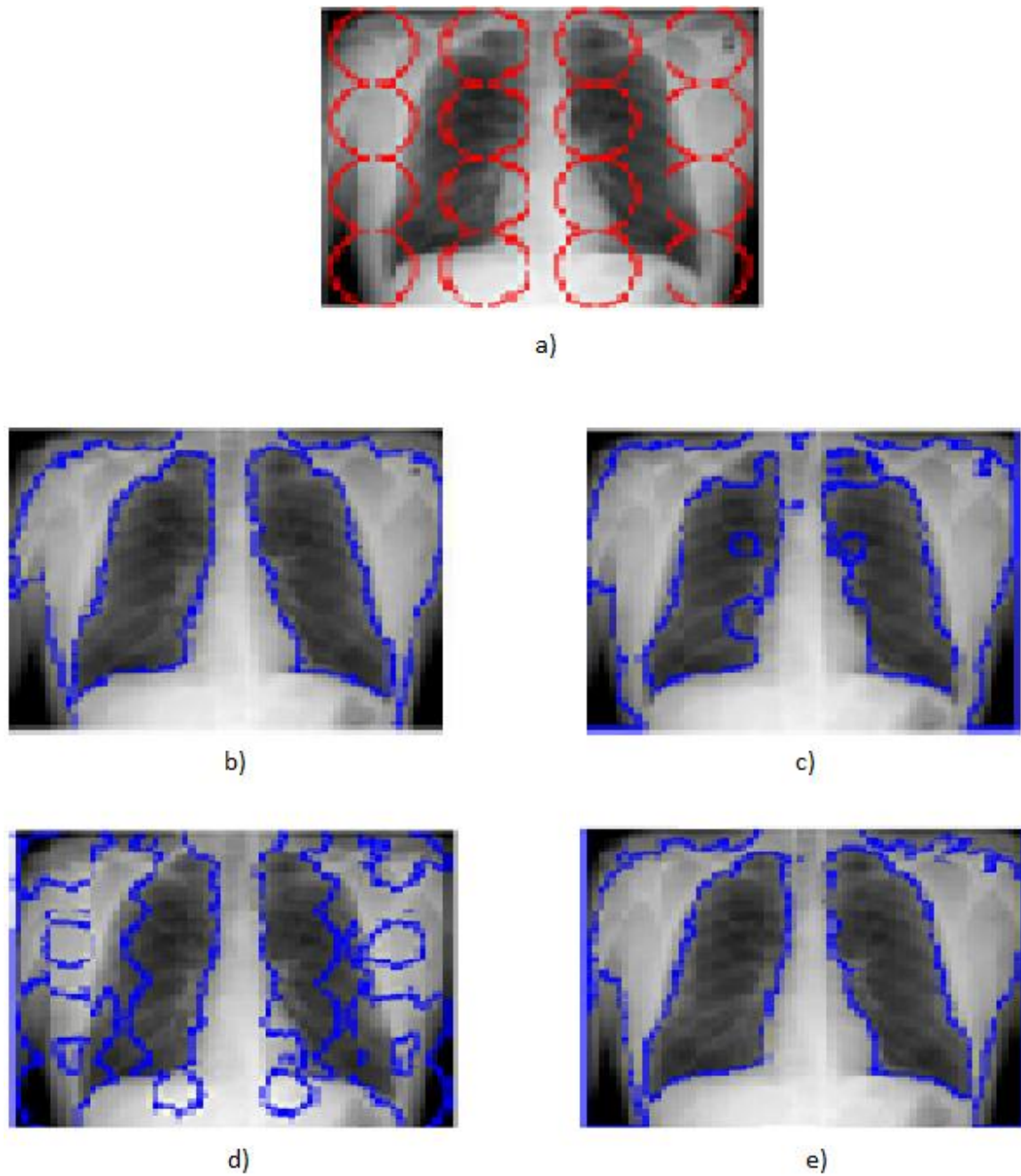
Dans cette partie nous allons faire des tests d'évaluation pour l'approche adoptée en la comparant avec d'autres méthodes (Chan et Vese, [23] et [22]). Les tests vont être effectués sur 5 images en niveau gris dont 2 bruitées. Notre choix c'est porté sur l'image 3 (figure III.28), image 15 (figure III.29), image 7 (figure III.30), en ce qui concerne les images non bruitées. Pour les figures (III.31 et III.32), nous avons utilisé les images bruitées : image 13 et 14.



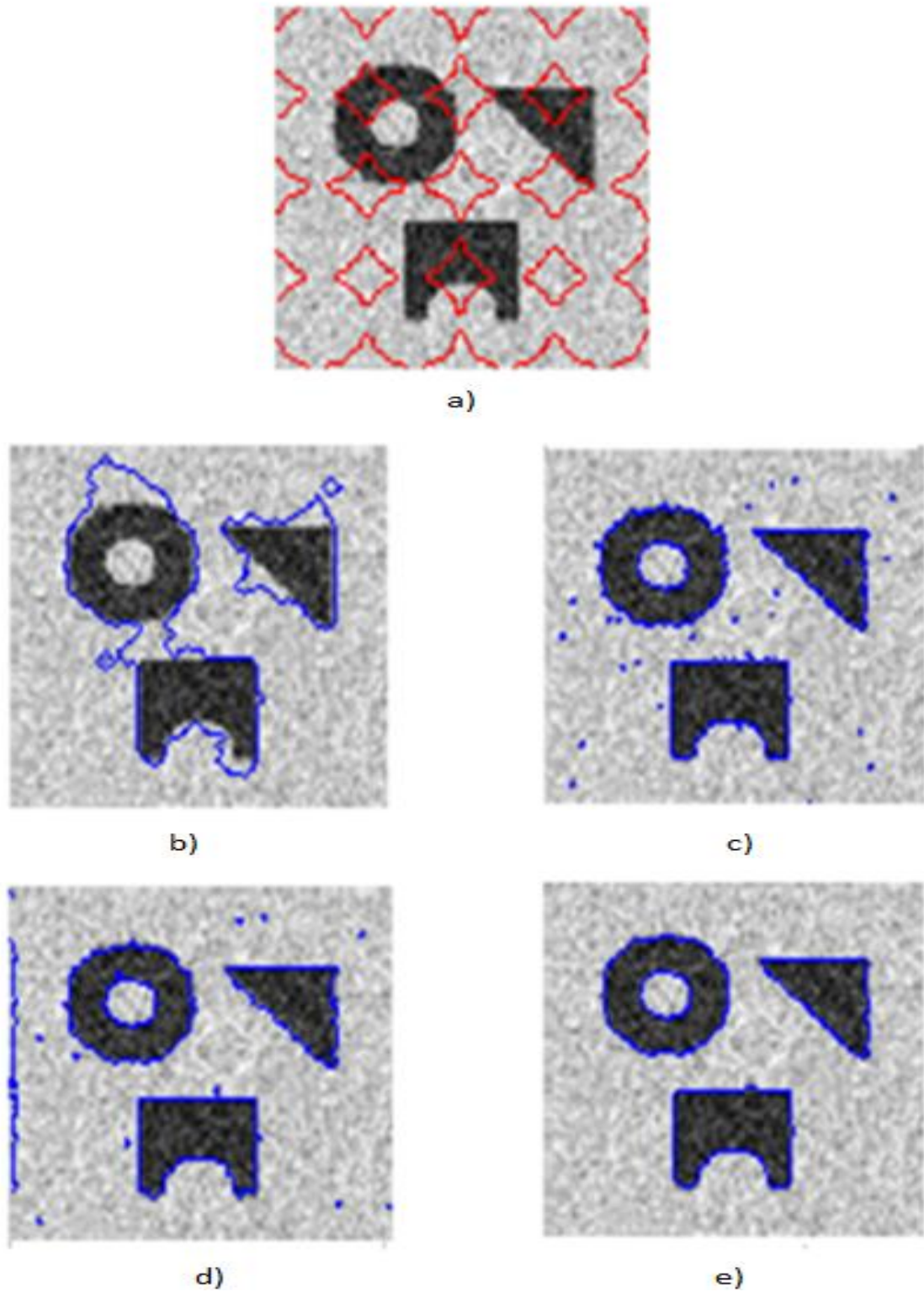
**Fig.III.28.** Résultat de segmentation : a)initialisation, b) approche adoptée, c)Chan et Vese, d) [23] et e) [22].



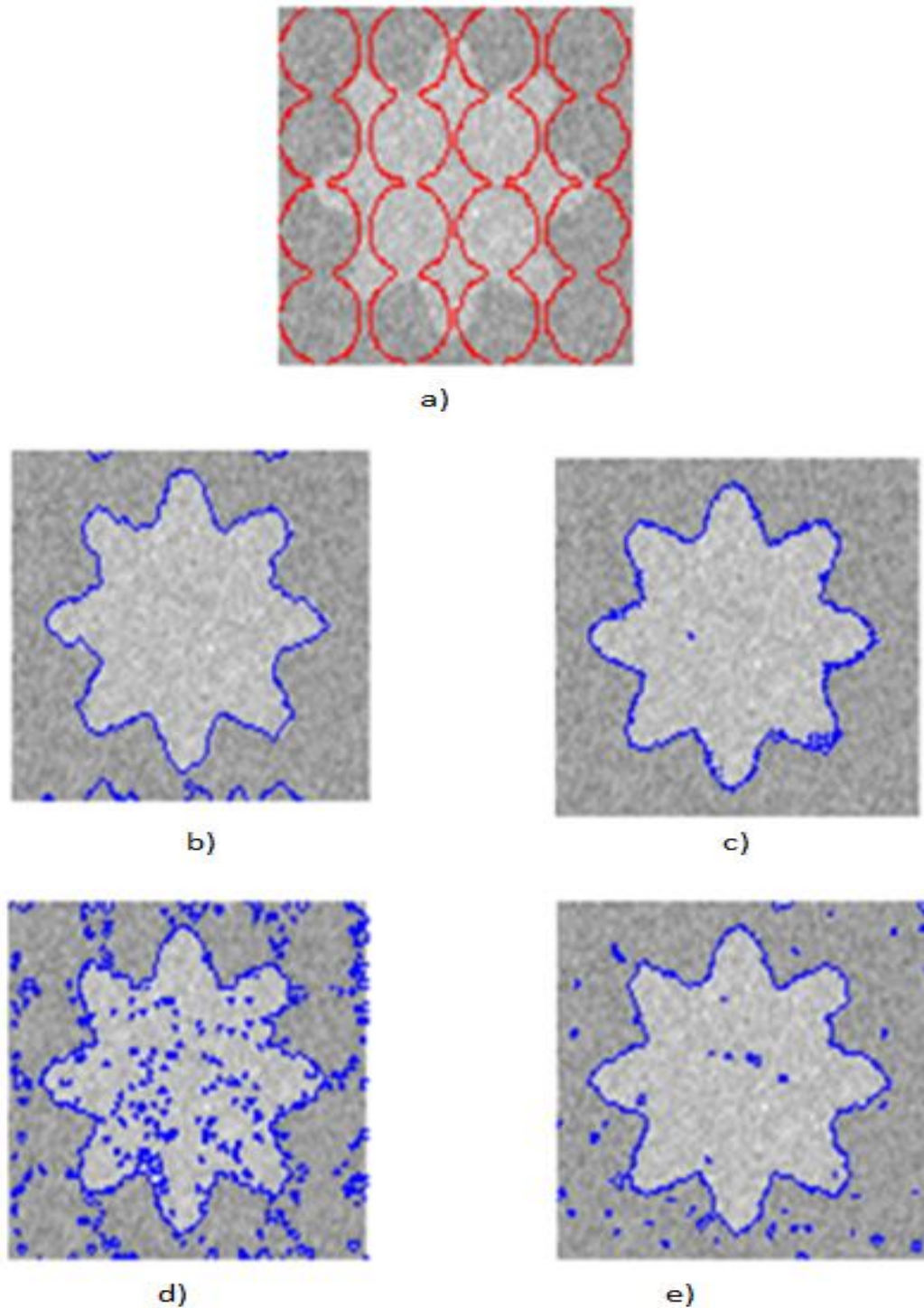
**Fig.III.29.** Résultat de segmentation : a)initialisation, b) approche adoptée, c)Chan et Vese, d) [23] et e) [22].



*Fig.III.30. Résultat de segmentation : a)initialisation, b) approche adoptée, c)Chan et Vese, d) [23] et e) [22].*



**Fig.III.31.** Résultat de segmentation : a)initialisation, b) approche adoptée, c)Chan et Vese, d) [23] et e) [22].



**Fig.III.32.** Résultat de segmentation : a)initialisation, b) approche adoptée, c)Chan et Vese, d) [23] et e) [22].

Nous avons dressé un tableau récapitulatif (tab III.1), où nous avons mentionné le temps et le nombre d'itération qu'a pris chaque méthode et sur chaque image. Dans ce tableau les chiffres en bleu représentent le temps de convergence le plus petit par contre en rouge le temps le plus grand.

		Image 3	Image 15	Image 7	Image 13	Image 14
temps(s)	Approche adoptée	1.7621	0.2575	34.8456	0.0801	0.0959
	Chan et Vese	9.0094	2.2302	40.5552	1.0969	1.3882
	[23]	1195.7362	57.3522	238.2182	23.1640	24.5360
	[22]	6.6520	0.9616	0.5203	0.0157	0.6366
Nombres des itérations	Approche adoptée	61	36	300	23	17
	Chan et Vese	300	30	300	300	300
	[23]	100	100	30	100	100
	[22]	300	300	6	4	300

**Tab III.1.** Tableau récapitulatif du temps et du nombre d'itérations.

Les résultats obtenus en termes de temps de segmentation pour l'image (3, 15, 14) dans la figure (III.28, III.29, III.32), respectivement, l'approche adoptée a le meilleur résultat (temps) secondée par l'approche [22] suivie par la méthode Chan et Vese, puis en dernier l'approche [23], pour l'images (7, 13) dans la figure (III.30, III.31), respectivement, le meilleur temps revient à l'approche [22] suivie par l'approche adoptée, puis Chan et Vese et approche [23]. Les résultats obtenus en termes de qualité de la segmentation, dans l'image 13 (bruitée) le résultat est moins bon, par contre satisfaisant pour les autres images.

Nous pouvons dire que l'approche adoptée réduit le temps de calcul considérablement mais reste tout de même la qualité de la segmentation moins bonne sur les images bruitées.

### III.5 conclusion

Dans ce chapitre nous avons présenté les différents tests que nous avons effectués sur des images différentes afin d'avoir une idée précise sur l'approche adoptée.

Nous pouvons conclure que l'approche adoptée a donné des résultats très encourageants du point de vue rapidité et qualité de segmentation en ce qui concerne les images en niveaux de gris, images en couleurs et les images 3D, vu les résultats obtenus.

Il reste tout de même que l'approche adoptée est sensible aux bruits et les résultats obtenus sur les images bruitées le justifient parfaitement. Aussi, une très bonne qualité de segmentation exige une initialisation très proche de l'objet.

## *Conclusion générale*

---

Le but de notre travail est de trouver une approche rapide de segmentation d'images par le modèle de contour actif.

Pour cela, nous avons fait une étude plus au moins détaillée sur les contours actifs et énoncé les différentes méthodes qui permettent d'accélérer le processus de convergence. Nous avons, par la suite adoptée une méthode, que nous avons développé en chapitre deux.

Ensuite, nous avons testé la méthode adoptée sur des images différentes, allant des images en niveaux de gris jusqu'aux images 3D, en passant par les images en couleur. Plusieurs initialisations ont été testées. Finalement un test comparatif avec d'autres méthodes d'accélération est effectué. Les résultats étaient assez encourageants, pour un modèle assez simple dans sa structure.

Cependant, nous pourrions conclure que c'est un modèle qui évolue localement et qui nécessite, de ce fait, une initialisation très proche des objets à détecter. Un prétraitement des images est, aussi, important pour éviter les points isolés.

Enfin, nous espérons que ce simple travail pourra servir à d'autres finalités et sera bénéfique pour les promotions à venir.

## Bibliographie

---

- [1] Sethian. J. ,” Fast Marching Level Set Method for Monotonically Advancing Fronts. Proc. Nat. Acad. Sei., 93, 4, pp.1591-1595, (1996)
- [2] Jean Jacques Rousselle, « Contours Actifs, une méthode de segmentation : Application à l'image médicale», Thèse de doctorat, université de tour, France, 2003.
- [3] S. Osher et J.A. Sethian, « Fronts propagating with curvature-dependent speed : Algorithms based on hamilton-jacobi formulation», Journal of Computational Physics, 79 : 12 - 49, 1988.
- [4] Ariane HERBULOT, «Mesures statistiques non-paramétriques pour la segmentation d'images et de vidéos et minimisation par contours actifs », Thèse de Doctorat, Université de Nice-Sophia Antipolis, France, 2007.
- [5] J.A. Sethian, “Level Set Methods”, Cambridge Univ. Press, 1996.
- [6] M. Kass, A. Witkin, et D. Terzopoulos, « Snakes : Active contour models », International Journal of Computer Vision, 1 : 321- 332, 1988.
- [7] S. Menet, P. Saint Marc, et G. Medioni. B-snakes, “Implementation and application to stereo”, Dans Image Understanding Workshop, pages 7207- 26, 1990.
- [8] L. Cohen. “On active contour models and balloons”. Computer Vision, Graphics and Image Processing : Image Understanding :211\_218, mars 1991.
- [9] D. Chopp, “Computing minimal surfaces via level set curvature”, Rapport technique, Lawrence Berkeley Laboratory, University of Berkeley, CA, 1991.
- [10] L. Cohen, E. Bardinet, et N. Ayache, « Surface reconstruction using active contour models”, SPIE Conference on Geometric Methods in Computer Vision, San Diego, CA, 1993.
- [11] M. Unser, “ Splines : A perfect \_t for signal and image processing”, IEEE Signal Processing Magazine16(6) :22-38, novembre 1999.
- [12] M. Jacob, T. Blu, et M. Unser, “ A unifying approach and interface for spline-based snakes”, Proceedings of the SPIE International Symposium on Medical Imaging : Image Processing, San Diego CA, USA, février 2001.

## Bibliographie

---

- [13] F. Precioso, M. Barlaud, T. Blu, et M. Unser, “Robust real-time segmentation of images and videos using a smooth-spline snake-based algorithm”, journal : IEEE Transactions on Image Processing, USA, juillet 2005.
- [14] V. Caselles, R. Kimmel, et G. Sapiro, “Geodesic active contours”, International Journal of Computer Vision, 22(1) :61-79, 1997.
- [15] James, Rathi, Yazzi, tannenbaum, “ Fast Approximate Surface Evolution in Arbitrary Dimension” , IS&T /SPIE symposium on Medical Imaging, 2008
- [16] V. Caselles, F. Catte, T. Coll, et F. Dibos, “A geometric model for active contours”, Numerische Mathematik, 66 :1-31, 1993.
- [17] R. Malladi, J.A. Sethian, et B.C Vemuri, “Shape modeling with front propagation : a level set approach”, IEEE Transactions on Pattern Analysis and Machine Intelligence, février 1995.
- [18] R. Ronfard, “Region-based strategies for active contour models”, International Journal of Computer Vision, 13(2) :229-251, 1994.
- [19] S. Zhu et A. Yuille, “Region competition : unifying snakes, region growing, and bayes/MDL for multiband image segmentation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 : 884-900, septembre 1996.
- [20] J.A.Sethian, “ *level set Methods and Fast Marching Methods*”, *Second Edition*, Springer New York, USA, 1999
- [21] T. Chan et L. Vese, “Active contours without edges”, IEEE Transactions on Image Processing, 10(2) :266-277, 2001.
- [22] Samir HACHOUR, Idir FEDDAG et Youcef IABADENE, « **Le suivi d’objets en mouvement par un modèle de contour actif** », Mémoire d’ingénieur, département Automatique, Université Mouloud MAMMERY, Tizi-Ouzou, Algérie, 2010.
- [23] Ahcen Ait-Menguellet, Arezki Slimani et Hamid Meziani, « Segmentation d’images médicales 3D par un contour actif rapide », Mémoire d’Ingénieur, Département d’Automatique, Université Mouloud MAMMERY, Tizi-ouzou,
- [24] J. Montagnat, H. Delingette, et G. Malandain, « Cylindrical Echocardiographic Image Segmentation Based on 3D Deformable Models », Dans International Conference on Medical Image Computing and Computer Assisted Intervention, LNCS, pages 168-175, Cambridge, UK, septembre 1999.

## Bibliographie

---

- [25] R. Malladi, J.A. Sethian, et B.C. Vemuri, “A fast level set based algorithm for topology independent shape modeling”, *Journal of Mathematical Imaging and Vision*, Special issue on Topology and Geometry, 6 :269 -289, 1996
- [26] S.W K. Zhang, L. Zhang, H. Song, and W. Zhou, “Active contour with selective local or global segmentation: a new formulation and level set method” *Image and Vis Comput*, vol. 28, no. 4, pp. 668-676, April 2010.
- [27] : S.Yonggang et W.C.Karl, « *A fast implementation of the level set method without solving partial differential equations* », Technical Report No. ECE-2005-02, Department of Electrical and Computer Engineering, university of Boston, 2005
- [28]. S. Yui, K. Hara, H. Zha, and T. Hasegawa, “A Fast Narrow Band Method and Its Application in Topology-Adaptive 3-D Modeling”. In *Proceedings of ICPR (4)*. 2002
- [29] N. Paragios et R. Deriche, “*Geodesic active contours and level sets for the detection and tracking of moving objects*”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3) :266–230, 2000.
- [30]Y. Shi et W. Karl, « A fast level set method without solving PDEs”, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, USA, 2005
- [31] Y. Shi et W. Karl, « Real-time tracking using level sets”, *IEEE Computer Vision and Pattern Recognition (CVPR)*, San Diego, USA, 2005.
- [32] J. A. Sethian. (1995) “A Fast Marching Level Set Method for Monotonically Advancing Fronts”, *Proceedings of the National Academy of Sciences*, 93, 4, 1591-1595, (1995).