الجمهوريةالجزائريةالديمقر اطيةالشعبية République Algérienne démocratique et populaire

وزارةالتعليمالـعـاليوالبحـثّالعـلمـي Ministère de l'enseignement supérieur et de la recherche scientifique

> جــامعةسعددحلبالبليدة Université SAAD DAHLAB de BLIDA

> > كليةالتكنولوجيا Faculté de Technologie

قسمالإلكترونيڭ Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

Oukil Abdelkader

&

Talbi Farid

pour l'obtention du diplôme master2 en Électronique option traitement d'information et système électronique.

Thème

# Implémentation sur FPGA d'une méthode de segmentation des images par Level Set

Proposé par : M<sup>me</sup>. F. ALIM-FERHAT & M<sup>r</sup> A.GUESSOUM

Année Universitaire 2011-2012

C'est avec grand plaisir que nous réservons cette page, en signe de gratitude et de reconnaissance à tous ceux qui nous ont aidés à la réalisation de notre projet.

Nous remercions Dieu, le tout puissant de nous avoir gardé en bonne santé, de nous avoir donné le courage et la patience pour l'accomplissement de ce projet.

Nous tenons à remercier M<sup>me</sup> F. ALIM FERHAT de nous avoir encadrées, pour son énorme effort fourni, sa confiance, son dévouement constant et son entière disponibilité. Qu'elle trouve ici l'expression de notre gratitude pour ses précieux conseils et toute l'aide qu'elle nous a procurée durant l'élaboration de ce projet.

Je remercie notre Co-promoteur **M** A.GUESSOUM pour ses conseils et orientations tant précieuse qu'il nous a prodigués durant ce projet. Ainsi que **M** L.ABDELLI. Que tous les membres de l'équipe AC2 (Architecture pour la Compression et la Cryptographie) de la division ASM (Architecture des Systèmes et Multimédia), trouvent ici nos remerciements, pour leurs aides et leurs conseils. Nous exprimons notre profonde gratitude à nos enseignants du département d'électronique de l'US D B.

Que le membre du jury trouve ici le témoignage de notre reconnaissance pour avoir bien voulu juger notre projet.

Enfin nos derniers remerciements, vont à nos parents, à nos sœurs, à nos frères, à nos nièces, à nos neveux pour leur soutien indéfectible.

## Dédicaces

Je dédie ce modeste travail à : A ma très chère mère, puisse dieu me la garder. A la mémoire mon défunt père et mon défunt frère. A ma chère sœur : Hannia. A mes chers frères : Mustapha, Ayache et Samir A mes neveux : Ahmed, Bilel. A mes amis : Rédha, Abdelrahim, Sidahmed et Abdelkader A mes voisins : Samir, Hakim, Djamel, Kheir-eddine et les autres. Avec l'expression de tous mes sentiments de respect, Je dédie ce modeste travail.

Farid.

## Dédicaces

Je dédie ce modeste travail à mes parents, qui m'ont encouragé, aidé et soutenu.

*Comme je le dédie à:* 

- Mes grandes mères : FATEMA EL ZOHRA et MALIKA.
- Mon frères et mes sœurs: MOHAMED, NASSIMA et AMEL.
- Ma fiancée : SOUMIA.
- Mes nièces : LINA et MANEL.
- Mes amis intimes : D.SADEK, Z.ABDELRAZEK, T.NADJIB, T.FARID, L.AMINE.
- et à tous ceux qui me sont chers.
- Avec l'expression de tous mes sentiments de respect,
- Je dédie ce modeste travail.

Abdelkader.

Les images font partie intégrante de la vie humaine, l'image s'intègre naturellement dans notre environnement quotidien, nous synthétisons des images à des fins artistiques ou autres. Nous générons et interprétons des images à des fins scientifiques comme les images médicales.

Dans le domaine médical, l'utilisation des images est à présent considérée comme indispensable pour des raisons de faciliter le diagnostic et aussi pour disposer des techniques d'analyses des images automatiques ou semi-automatiques grâce à des systèmes informatiques.

Ces systèmes permettent de proposer aux praticiens une nouvelle façon de travailler en obtenant de nouvelles mesures par des techniques quantitatives et objectives. Le challenge est de taille, la réalisation de systèmes informatiques aidant le médecin doit permettre de sécuriser le diagnostic, le rendant ainsi plus fiable.

La première étape dans le processus de compréhension d'une image est le plus souvent l'extraction des objets qui la composent. La segmentation représente une étape importante dans le traitement d'images, puisqu'elle se situe à la charnière entre l'acquisition des images et l'utilisation de la sémantique qu'elles contiennent. La segmentation d'image est une opération de bas niveau qui consiste à localiser dans une image les régions appartenant à une même structure (objet). Tout cela débouche souvent sur des problèmes de reconnaissance de formes : reconnaissance de visages d'empreintes digitales, d'iris de l'œil, de textes, de dessins. De nombreuses recherches ont eu lieu sur les méthodes de segmentation, il en résulte un très grand nombre de méthodes. Aussi, la segmentation est primordiale en analyse d'images, du simple seuillage des niveaux de gris aux techniques plus complexes comme les contours actifs géométriques (Level Set). Nous nous sommes intéressés à la méthode des modèles déformables, c'est une technique particulière de segmentation. Son principe est de faire évoluer un contour initial selon certaines lois pour aboutir aux bords de l'objet à segmenter, ce qui permet d'avoir un contour fermé et mince. En effet les points de contour constituent rarement des courbes fermées et une étape de fermeture est nécessaire.

Le mémoire est organisé de la manière suivante :

Dans le premier chapitre, nous commençons par une étude des principales techniques de traitement d'images ainsi que les différents blocs qui constituent un système de traitement d'image. Deux approches ont été abordés concernant le bloc de segmentation: l'approche région et l'approche frontière.

Dans le deuxième chapitre, nous présentons les différentes méthodes de segmentation par les contours actifs, les contours actifs paramétriques et leurs inconvénients, par exemple, leur sensibilité à la position et à la forme du contour initial, ainsi que leur difficulté de gérer automatiquement le changement de topologie de la courbe en cours d'évolution. Ensuite, les différentes méthodes appartenant à la classe géométrique, qui sont respectivement: La méthode par ensemble des niveaux sans réinitialisation et La méthode par ensemble des niveaux avec réinitialisation de la fonction de distance.

Dans le troisième chapitre, nous présentons les résultats obtenus par l'application d'algorithme d'ensemble des niveaux sans réinitialisation sur les déférents types d'images (Synthétique, IRM). La validation de notre algorithme est assurée par l'outil MATLAB.

Dans le dernier chapitre, nous commençons par la présentation de l'outil ISE 7.1 de Xilinx utilisé pour l'implémentation de notre architecture. Les résultats de simulation et de synthèse des différents blocs de notre architecture ainsi que les résultats de placement et routage sur le circuit FPGA sont donnés à la fin du chapitre. Nous terminons ce mémoire par une conclusion générale et des perspectives.

#### ملخص

Field ) FPGA مالهدف الرئيسي من هذا العمل هو زرع في الدارة المنطقية المبرمجة FPGA ( FPGA الهدف المبرمجة Field ) FPGA تقنية بنطلاقا من صورة ، مستخدما تقنية بدون تجديد مجموعة المستويات .

في المرحلة الأولى تم إنجاز خوارزمية هذه الطريقة وبرمجتها من طرف برنامج Matlab في المرحلة الأولى تم إنجاز خوارزمية هذه الطريقة وبرمجتها من طرف برنامج ISE 7.2.

كلمات المفاتيح: معالجة الصور, تقسيم,مجموعة المستويات,VHDL, FPGA.

## Résumé

L'objectif essentiel de notre mémoire est l'implémentation sur un circuit FPGA (Field Programmable Gate Array) d'une méthode de segmentation des images par ensemble des niveaux sans réinitialisation. Dans un premier temps, un algorithme a été élaboré et validé par l'outil Matlab. Par la suite, une architecture a été proposée et implémentée sous l'environnement Xilinx ISE 7.2. Mots clés : Traitement d'image, segmentation, ensemble des niveaux,

VHDL, FPGA.

## Abstract

The main objective of our work, is the implementation of an image segmentation on FPGA (Field Programmable Gate Array) circuit by using a Level-Set without re-initialization method. At first, an algorithm has been elaborate and tested under Matlab software. Next, an architecture has been proposed and implemented under Xilinx ISE 7.2 environment.

Key words: Image processing, segmentation, level set, VHDL, FPGA.

## Chapitre 1 Généralités sur traitement d'image et

## la segmentation

## 1.1 Introduction

L'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises. De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

## 1.2 Définition de l'image

L'image est une représentation d'une sciène ou d'un objet par la peinture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction I(x, y) de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation [1].

## 1.3 L'image numérique

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits).

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy). En une image numérique représentée par une matrice bidimensionnelle de valeurs numériques f(x, y) où (x, y) les coordonnées cartésiennes d'un point de l'image, f(x, y) est le niveau de gris en ce point pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de tableau I de n lignes et p colonnes. Chaque élément I (x, y) représente un pixel de l'image et à sa valeur est associé un niveau de gris codé sur m bits (2<sup>m</sup> niveaux de gris, 0 = noir, 2<sup>m</sup>-1 = blanc). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur [1].

## 1.4 Les caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

#### 1.4.1 Pixel

Contraction de l'expression anglaise " Picture elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :



Figure 1.1. Le pixel.

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

## 1.4.2 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

## 1.4.3 Bruit

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

## 1.4.4 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

#### **1.4.5** Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

#### 1.4.6 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

1. Des images lumineuses (brillantes).

2. Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.

3. L'absence de parasites.

#### 1.4.7 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

4

Si L<sub>1</sub> et L<sub>2</sub> sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$
(1.1)

#### 1.4.8 Image à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

#### 1.4.9 Image en couleurs

Parfois, il est utile de représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, etc.) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.).

## 1.5 Les systèmes de traitement d'images



Un système de traitement numérique d'images est composé de :

Figure 1.2. Système de traitement numérique d'images.

## 1.5.1 Acquisition des données de l'image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de l'image d'origine à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales : les caméras numériques et les scanners. Notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc.

Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V.

#### a Dispositifs de numérisation d'images

Suivant l'objet ou le document à numériser et le domaine d'application dans lequel l'image va être utilisée, il existe divers dispositifs de numérisation d'images allant du simple scanner à main au satellite de télédétection.

#### 1.5.2 Le prétraitement

Avant de traiter une image il est nécessaire d'en définir la qualité. Un examen visuel permet de nous orienter sur la question. Cependant, pour une analyse objective de la qualité, il existe des outils statistiques (l'histogramme par exemple), qui permettent une analyse globale de l'image à partir de laquelle nous pouvons effectuer des modifications sur celle ci.

L'étape de prétraitement permet d'améliorer la qualité d'une image. C'est une étape importante dans une chaîne de traitement d'images. Pour la segmentation, elle permet d'éliminer une partie du bruit et, de mettre en valeur les contours. Pour pallier à ces dégradations on utilise en général le filtrage [2].

#### a Filtrage numérique

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage. Le filtrage consiste à modifier la distribution fréquentielle des composantes d'un signal selon des spécifications données. Le système linéaire utilisé est appelé filtre numérique [3].

Parmi ces systèmes, nous distinguons : les filtres passe-bas (lissage), filtres passe-haut (accentuation), filtres passe-bande (différenciation) et filtres directionnels.

#### **1.6** La segmentation

La segmentation est un traitement de bas niveaux qui consiste à créer une partition de l'image observée en un certains nombres de régions qui pourront par la suite être traitées de manière différente [4]. Il n'y a pas de méthode unique de segmentation d'une image. Le choix d'une technique est lié à plusieurs facteurs comme : la nature de l'image (optique, infrarouge, etc.), les conditions d'acquisition (éclairage, présence de bruit, etc.) les primitives à extraire (contours, segments de droite, textures, etc.). Du fait de cette diversité, il est difficile de définir de manière absolue une bonne segmentation.

7

La segmentation fait souvent référence aux notions de différence et de similarité comme les perçoit le système visuel humain ; et ceci donne naissance à deux approches couramment qualifiées d'approche contours et d'approche régions.

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

$$\forall \mathbf{i} \ \mathbf{R}_{\mathbf{i}} \neq \emptyset \quad \forall \mathbf{i}, \mathbf{j} \ \mathbf{i} \neq \mathbf{j} \ \mathbf{R}_{\mathbf{i}} \cap \mathbf{R}_{\mathbf{j}} = \emptyset \quad \mathbf{I} = \bigcup_{\mathbf{i}} \mathbf{R}_{\mathbf{i}}$$
(1.2)

#### 1.6.1 Approche région

La segmentation par région vise à rassembler dans une même région tous les pixels connexes qui partagent les mêmes propriétés (texture, couleur) [5]. De nombreuses méthodes de segmentation par région ont été développées ces trente (30) dernières années, nous présentons ci-dessous quelques méthodes de segmentations par régions.

#### a segmentation par classification

Cette méthode détermine d'abord une partition de l'espace des luminances en utilisant les niveaux de gris présents dans l'image. On associe à chaque pixel la classe de niveaux de gris à laquelle il appartient. Les régions sont définies par les ensembles maximaux de pixels connexes appartenant à la même classe.

Le plus souvent la classification des luminances s'effectué à partir du calcul d'histogramme de répartition dans l'image. On recherche les différents modes (maxima) de l'histogramme et les vallées (minima) correspondantes. Les classes sont déterminées par les intervalles entre les vallées .cette procédure fonctionne bien pour des images comprenant un nombre peu important d'objets ayant des niveaux d'intensité différents [4].

8



Figure 1.3. Histogramme de répartition pour trois classes.

#### Exemples

 La classification non supervisée: elle consiste à découper l'espace de représentation en zones homogènes selon un critère de ressemblance entre les individus [6], c'est à dire selon un critère de proximité dans l'espace des attributs.

2. La classification supervisée: ou classification avec apprentissage consiste à construire, à partir d'une classification connue a priori d'un certain nombre d'individus, une fonction d'identification ou de discrimination pour les autres individus. Cette fonction discriminante est construite suivant des méthodes établies.

#### b segmentation par division (split)

Cette méthode consiste à diviser l'image, qui constitue la région initiale, en régions de plus en plus homogènes .Le processus est réitéré pour chacune des régions produites jusqu'à ce qu'une certaine homogénéité soit atteinte. L'homogénéité d'une région est souvent contrôlée par sa variance ou son contraste. Ces techniques à caractère descendant ont une faiblesse liée à la nature souvent régulière du découpage. Une région est divisée en sous régions de niveaux inférieurs, les frontières d'une région sont alors représentées sur différents niveaux ; leurs délimitations exactes sont ainsi difficiles à obtenir. Beaucoup d'algorithmes de division reposent sur l'utilisation des histogrammes de niveaux de gris .Les régions sont alors définies à partir des intervalles entre les vallées [7].



Figure 1.4. Segmentation par division.

#### c approche fusion

L'idée consiste à exploiter une partition initiale de l'image constituée de petites régions. Puis ces régions sont fusionnées successivement jusqu'à ce que le critère de fusion ne soit plus vérifié. Plusieurs règles de regroupement ont été proposées. Certaines de ces règles mettent en jeu:

1. Des propriétés statistiques telles que la moyenne ou la variance des niveaux de gris des régions, le gradient moyen des frontières de régions, le contraste maximum des régions, ou d'autres statistiques locales qui expriment l'état de surface des régions, etc.

 Des propriétés géométriques ou morphologiques telle que l'élongation ou la compacité des régions. Deux régions sont regroupées si par exemple un facteur de forme est conservé ou amélioré après leur fusion [7].







Figure 1.5. Segmentation par division.

#### d approche par division-fusion

Ces méthodes combinent les deux approches précédentes : la division qui partitionne l'image en zones localement homogènes, puis la fusion des régions similaires au sens d'un prédicat de regroupement. Ces deux opérations sont répétées jusqu'à ce qu'elles ne soient plus possibles.



Partitions

fusion

Figure 1.6. Principe de la division-fusion.

## **1.6.2** Approche frontière

#### a introduction

Dans n'importe qu'elle image, il existe des variations d'intensité représentant des changements des propriétés physiques ou géométriques de la scène.

La segmentation est basée sur ces variations, car ces dernières peuvent être interprétées par des informations très importantes. Elles constituent les frontières des régions correspondant à des bords ou parties d'objets de la scène. D'où le nom donné à cette approche de la segmentation.

La détection de contours peut être perçue différemment selon le but recherché. Ainsi, selon Pavlidis [7], elle consiste en une détection des discontinuités présentes dans une image. La détection des contours est alors considérée comme l'une des premières étapes de la vision par ordinateur et aucune information a priori sur les objets présents dans l'image n'est utilisée. Toutefois, dans certaines applications, il est utile d'exploiter des informations sur la forme et l'emplacement approximatif de l'objet pour détecter le contour.

Par définition, un contour est une brusque variation du niveau de gris dans une image d'une amplitude a et avec une pente p. Un contour peut être défini comme une "marche d'escalier" si le contour est net, comme une "rampe" si le contour est plus flou ou comme un "toit" s'il s'agit d'une ligne sur un fond uniforme [8].



Figure 1.7. Quelques profils de contour (a) : marche, (b) : rampe, (c) : toit.

Pour déterminer la position des éventuels contours, on utilise des méthodes dérivatives. En effet, l'image dérivée met en évidence les variations brusques de niveaux de gris. Quand il y a une forte variation, cela est souvent dû à un changement de teinte d'éclairement ou de texture et de façon générale, à un changement de zone.

Les premières idées de modélisation d'une discontinuité étaient très simples. Une discontinuité était assimilée à un fort gradient ou à un passage par zéro du Laplacien.

Dans ce paragraphe nous présentons, trois méthodes fondamentales de la segmentation pour la détection des contours, qui sont :

1. Les méthodes dérivatives : Ce sont les plus immédiates pour détecter et localiser les variations du signal [9]. Les contours sont assimilés aux points de fort gradient ou de dérivée seconde nulle.

2. Les méthodes morphologiques : Elles travaillent sur les maximums et les minimums des intensités du voisinage de chaque pixel. Un contour de type « saut d'amplitude » sera détecté si la différence entre le maximum et le minimum est importante.

3. Les contours actifs: Le principe des contours actifs est de faire évaluer un contour initial autour de l'objet d'intérêt vers une position d'équilibre.

#### b les méthodes dérivatives

Le principe général des méthodes dérivatives est illustré sur un signal continu monodimensionnel A(x) présentant une transition avec un saut d'amplitude en  $x_0$  [9]. Les allures de la dérivée première et de la dérivée seconde sont données par la

figure 1.8 (où x<sub>0</sub>=0).

Si on considère que la transition du signal est repérée par son point d'inflexion, sa localisation peut se faire par recherche du maximum local de la valeur absolue de la dérivée première ou par recherche du passage par zéro de la dérivée seconde.



*Figure 1.8.* Allure des dérivées premières et seconde d'une transition de type 'Saut d'amplitude'.

Dans le cas bidimensionnel, l'image est représentée par une fonction scalaire continue A(x, y). Le vecteur gradient est défini au point M de coordonnées (x, y) par :

$$\nabla A(x, y) = \left(\frac{\partial A}{\partial x} \quad \frac{\partial A}{\partial y}\right)^{t}$$
 (1.3)

Le vecteur gradient en un point M est normal à la courbe de niveau donné par A(x, y)=Constante qui passe par M. La plus grande variation de A(x, y) a lieu quand on se déplace le long de la normale à la courbe de niveau. Le maximum de cette variation est défini en grandeur et en direction par  $\nabla$  A(x,y).

Au point M(x, y), l'orientation  $\phi$  du gradient est donnée par :

$$\nabla A(x,y) = \arctan\left(\frac{\partial A}{\partial y}/\frac{\partial A}{\partial x}\right)$$
 (1.4)

Le vecteur unitaire normal au contour « n » et l'orientation du contour «  $\theta$  » ont pour expression :

n = (cos
$$\phi$$
 sin $\phi$ ) et  $\theta = \frac{\pi}{2} - \phi$  (1.5)



*Figure 1.9.* Orientation du contour.

#### b.1 opérateurs dérivatifs du premier ordre

Ces opérateurs peuvent être appliqués sur les images numériques «cas discret », les dérivées directionnelles suivant les directions horizontales et verticales au site [i, j] sont approchées par de simples différences finies [7]:

$$\frac{\partial A}{\partial y} \approx \frac{\Delta A}{\Delta i} = A_i[i, j] = A[i+1, j] - A[i, j]$$
(1.6)

$$\frac{\partial A}{\partial x} \approx \frac{\Delta A}{\Delta j} = A_{j}[i, j] = A[i, j+1] - A[i, j]$$
(1.7)

La norme du gradient est donnée par :

$$|\nabla A[i, j]| = \sqrt{A_j^2[i, j] + A_i^2[i, j]}$$
 (1.8)

Ou encore :

$$|\nabla A[i, j]| = MAX \{ |A_j[i, j]|, A_i[i, j] \}$$
 (1.9)

Ou encore :

$$|\nabla A[i, j]| = \frac{\{A_j[i, j] + A_i[i, j]\}}{2}$$
 (1.10)

*Remarque* En pratique la majorité des opérateurs sont implémentés sous forme de masque.

#### b.1.1 operateurs de prewitt et de sobel

Pour ces opérateurs, les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$A_i[i, j] = h_i^* A[i, j]$$
 et  $A_j[i, j] = h_j^* A[i, j]$  (1.11)

avec :

$$h_{j} = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \quad \text{et} \quad h_{i} = \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix}$$

Les matrices h<sub>i</sub> et h<sub>j</sub>, appelées aussi masques, sont les noyaux de convolution de filtres à réponse impulsionnelle finie.

Les masques de Prewitt [10] sont définis par c=1 et les masques de Sobel par c=2.

Rappelons que l'opération de convolution peut être traduite par l'expression:

• •

h \* A [i, j] = 
$$\sum_{m=-M}^{M} \sum_{n=-N}^{N} h(m, n) \cdot A[i - m, j - n]$$
 (1.12)

Où h est le noyau de convolution de taille (2M+1)\*(2N+1).

...

Les calculs précédents des deux dérivées directionnelles peuvent donc être considérées comme des corrélations avec les « gabarits »  $h_i$  [-j] et  $h_i$  [-i].

Exemple

Image Originale



Image Originale



Opérateur de Prewitt



Opérateur de Sobel



#### Figure 1.10. Applications des détecteurs de Prewitt et Sobel.

*Remarque* Nous remarquons que l'application de ces deux masques ne donne pas de bons contours et fait apparaître quelques points constituant un bruit. Il faut signaler que le masque de Sobel donne des contours accentués et épais.

#### b.1.2 Opérateurs de gradient directionnel de kirsh

L'opérateur de Kirsh [11] consiste à filtrer l'image avec 8 masques directionnels.

L'orientation du contour (orientation du filtre + 45°) est dans ce cas déduite par le filtre donnant le résultat le plus élevé (qui est considéré comme représentant l'intensité du gradient).

Exemple de tels filtres:

$$\begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}, \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}, \text{ etc.}$$

#### b.1.3 opérateurs MDIF

Il s'agit de la combinaison d'un filtre moyenneur du noyau m et d'un dérivateur utilisant les masques directionnels de Prewitt  $h_i$  et  $h_i$ :

$$\mathsf{m} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \ \mathsf{h}_i = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \ \mathsf{h}_j = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

Les approximations des composantes du gradient peuvent être calculées après le lissage ou directement en appliquant les masques suivants de taille 5x5

$$\mathbf{m}_{j} = \mathbf{m}^{*}\mathbf{h}_{j} = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{m}_{i} = \mathbf{m}^{*}\mathbf{h}_{i} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix}$$

Rappelons que « \* » est l'opération de convolution et les images sont obtenues par convolution de l'image originale A avec les deux masques  $m_i$  et  $m_j$ .



Figure 1.11. Opérateur MDIF.

La nature du filtre MDIF (lissage + détecteur dérivatif) fait apparaître en plus des contours une sorte de flou.

## b.1.4 opérateur NAGDIF

L'opérateur NAGDIF [9] est la combinaison d'un lissage non linéaire de Nagao suivi d'une opération de dérivation.

Le filtrage de Nagao est un filtrage par fractionnement de la fenêtre d'analyse. En effet, la sélection des pixels intervenant dans le calcul de la sortie du filtre est faite en prenant en compte des contraintes géométriques.

L'opérateur de Nagao travaille sur un voisinage 5x5 centré sur le pixel courant. Dans ce voisinage, 9 domaines sont définis. Pour chaque domaine Di, la moyenne  $\mu_i$  et la variance var( $\sigma_i$ ) sont calculées. L'intensité du pixel courant est remplacée par la moyenne du domaine qui présente la plus faible variance.

	 	· · · · · · · · · · · · · · · · · · ·	
		1	1
	1	1	1
( ) (A)		1	1

Les domaines D1, D2 et D4 sont obtenus par rotation de  $\pi/2$ .



Exemple

Les domaines D5, D6 et D8 sont obtenus par rotation de  $\pi/2$ .

1	1	1	
1	1	1	
1	1	1	

Domaine D9

Figure 1.12. Domaines de Nagao.

Cet opérateur étend une technique fondée sur un lissage non-linéaire avec des domaines carrés. Il présente l'avantage de réduire le bruit tout en renforçant les contrastes par diminution de la largeur de la zone de transition. Le domaine D9 peut être ignoré sans altérer notablement les performances.

La dérivation pour l'opérateur du NAGDIF est effectuée par l'opérateur suivant :

CONTOUR Si MAX {B [i, j] – B [i -  $\varepsilon_1$ , j -  $\varepsilon_2$ ] } > Seuil. (1.13)

Où  $\varepsilon_1$ , et  $\varepsilon_2$ sont des valeurs entières de l'intervalle [-1, 1], B étant l'image lissée par le

filtre de Nagao et la valeur de seuil est déterminée expérimentalement.

Exemple







Figure 1.13. Opérateur NAGDIF.

Les contours détectés sont minces du fait que les transitions dans l'image originale ont de petites largeurs.

#### b.2 operateurs dérivatifs du deuxième ordre

Les opérateurs dérivée seconde permettent de détecter de brusques variations d'intensité dans les images en sélectionnant les points dont la dérivée première présente une brusque variation d'intensité. Ils permettent de réduire les erreurs de fausses alarmes en rejetant les gradients de faible intensité. Les points de contours, qui correspondent à un maximum local dans le cas du gradient, correspondent à des passages à zéro du Laplacien [12], [9].

$$\Delta A(\mathbf{x}, \mathbf{y}) = \frac{\delta^2 A}{\delta \mathbf{x}^2} + \frac{\delta^2 A}{\delta \mathbf{y}^2}$$
(1.14)

En coordonnées polaires, cet opérateur s'exprime sous la forme :

$$\Delta A(r,\theta) = \frac{\delta^2 A}{\delta r^2} + \frac{1}{r} \frac{\delta A}{\delta r} + \frac{1}{r^2} \frac{\delta^2 A}{\delta \theta^2}$$
(1.15)

Nous remarquons ainsi qu'il est invariant par rotation.

En effet, tous les extremums du gradient correspondent à des zéros dans la dérivée de celui-ci. Pour ne pas être sensible aux valeurs minimales du gradient, c.-à-d. les valeurs nulles correspondant à des zones homogènes, il est nécessaire de détecter les passages par zéro du Laplacien et non pas seulement les valeurs nulles de celui-ci. Des approximations discrètes du Laplacien, calculées sur un voisinage 3x3, correspondent aux masques suivants [7]:

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} Ou \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Le problème majeur des techniques exploitant le Laplacien est la sensibilité aux petites variations non significatives dues essentiellement au bruit. En effet, un des problèmes fondamentaux de la détection des contours est la résistance au bruit et aux petites variations locales dues à la texture ou à des irrégularités de surface. Pour remédier à ce problème, il est possible d'effectuer un prétraitement sur l'image afin de lisser les petites irrégularités locales, Ce prétraitement peut être réalisé par des filtres de différentes natures.

Pour plus de précautions, il est aussi possible d'éviter les passages par zéro dans les zones avec variations locales faibles et irrégulières en acceptant uniquement les pixels dont la valeur du gradient est supérieure à un seuil prédéfini.



Figure 1.14. Opérateurs du deuxième ordre.

#### b.2.1 autres operateurs du deuxième ordre

Marr et Hildreth [9],[13] suggèrent par similitude avec l'œil humain, l'utilisation d'un opérateur Laplacien-Gaussien. La méthode proposée consiste à convoler l'image par une gaussienne avant d'appliquer un filtre Laplacien. Ils proposent l'utilisation de plusieurs masques correspondant à différentes valeurs de la variance  $\sigma^2$  de la gaussienne. Aussi, Huertas et Médioni [9], [13] proposent un filtre qui est une décomposition en filtres séparables de l'opérateur Laplacien d'une gaussienne.

Cependant, les masques de variance trop large dégradent l'image et ne permettent pas la détection des structures fines, à l'inverse, une variance trop faible conduit à une détection du bruit trop important. La localisation par ce type de méthodes n'est pas très précise et dans le cas d'images complexes, les résultats obtenus sont très sensibles au bruit et engendrent souvent de fausses détections. En effet, les faux contours sont détectés dans les zones bruitées ou texturées (même légèrement) et des contours significatifs dont le gradient est faible sont oubliés.

Pour surmonter ces difficultés de précision de localisation et d'efficacité de détection, canny [14] propose de formaliser mathématiquement une approche optimale de détection.

Pour déterminer le filtre optimal, il définit trois critères : bonne détection, bonne localisation et non multiplicité des maxima locaux [15].

Trois filtres optimaux majeurs ont été obtenus en prenant en compte ces trois critères : le filtre de Canny, le filtre de Deriche et le filtre de Shen et Castan.

Canny trouva la solution suivante, qui est la dérivation de la convolution d'une image avec une gaussienne. L'approche proposée sert toujours de référence dans la littérature car sa solution est correctement formalisée et le comportement du filtre est ainsi connu avec précision. Toutefois, certaines limites peuvent être recensées dans ce modèle : Ce type de filtrage permet de détecter correctement les contours, mais le fait d'augmenter la valeur de  $\sigma$  ne suffit pas pour détecter la totalité des contours. En effet, ils présentent quelques interruptions notamment au voisinage des coins. En plus, si l'image est dégradée par un bruit multiplicatif, ce qui est le cas des images obtenues par un système laser ou radar, le modèle de Canny est inadapté, puisque son modèle est proposé avec l'hypothèse que l'image est affectée avec un bruit blanc gaussien.

#### c La morphologie mathématique

La morphologie mathématique est une technique d'analyse de l'image basée sur la notion des ensembles. Les opérations de morphologie mathématique s'appliquent aux images binaires, mais elles peuvent aussi être appliquées aux images en niveaux de gris La morphologie mathématique comporte plusieurs opérateurs, chacun servant à atteindre un but précis.

#### c.1 Transformations morphologiques

#### c.1.1 Principes généraux

La morphologie mathématique est un ensemble d'outils inventé par Jean Serra et Georges Matheron [16] pour étudier et analyser des images et rechercher des éléments particuliers. Elle s'appuie sur des bases mathématiques dont la plus importante est la théorie des ensembles.

Il existe deux approches de la morphologie mathématique:

1. l'approche ensembliste, pour des images binaires.

2. l'approche fonctionnelle, pour des images en niveaux de gris.

Le principe général de la morphologie mathématique est d'analyser la structure d'une image en la comparant à un objet de référence appelé élément structurant. Les opérateurs de base de la morphologie mathématique sont la dilatation et l'érosion, tous les autres opérateurs étant construits à partir de ces deux derniers [17].

## c.1.2 Elément structurant

Un élément structurant, aussi appelé SE, est un ensemble utilisé pour appréhender la représentation topographique de l'image. Pour chaque SE, il faut définir sa taille, sa forme ainsi que son centre. Ces trois caractéristiques sont adaptées selon les informations de l'image devant être extraites [18].

L'élément structurant est un masque que nous faisons parcourir sur toute l'image. Les éléments structurants peuvent avoir des formes et tailles variées: cercle, disque, carré, segment, sphère, etc.

#### c.2 Opérateurs fondamentaux

#### c.2.1 La dilatation

Elle est représentée par la ligne externe à l'objet. Elle augmente la taille des particules, peut combler des concavités et connecter des particules.

#### c.2.1.1 Dilatation des images binaires

La dilatation d'un ensemble A par un ensemble B (l'élément structurant) forme un nouvel ensemble A' appelé dilatation de A par B. A' est l'ensemble des points décrits par l'élément structurant B lorsque son centre est contenu dans l'ensemble A.

Visuellement, le résultat de la dilatation est un agrandissement de la figure A et s'il existe des trous dans A, ceux-ci sont en partie comblés.

#### Exemple





*Figure 1.15*. Exemple de dilatation sur une image binaire.

C'est une transformation morphologique qui combine deux ensembles en utilisant l'addition vectorielle. Donc, si a et b sont deux sous ensembles de E, alors la dilatation de A par B notée A  $\bigoplus$  B est définie par:

$$A \bigoplus B = \{c \in E / (a \in A \text{ et } b \in B) c = a + b\}$$
 1.16

#### c.2.1.2 Dilatation des images à plusieurs niveaux de gris

Le dilaté d'un pixel est la valeur maximale au voisinage de ce pixel comprise dans l'élément structurant. La définition citée plus haut ne fonctionne que pour les images binaires, mais celle-ci peut s'appliquer également aux images à plusieurs niveaux de gris [19].

#### Exemple







Figure 1.16. Exemple de dilatation sur une image à plusieurs niveaux de gris

Les propriétés de la dilatation sont les suivantes:

c.2.2 L'érosion	
3. $A \subseteq B \Rightarrow A \bigoplus D \subseteq B \bigoplus D$	(1.19)
2. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$	(1.18)
1. $A \oplus B = B \oplus A$	(1.17)

L'érosion est l'opération duale de la dilatation, l'érosion d'un objet par un élément structurant est l'ensemble des points de l'objet pour lesquels l'élément structurant est totalement inclus dans l'objet.

L'érosion supprime les petites particules, réduit la taille des autres, supprime des pics et peut déconnecter certaines particules.

## c.2.2.1 Erosion des images binaires

L'érosion d'un ensemble A par un ensemble B est l'ensemble A' des points x tel que B translaté par x soit toujours inclus dans A. A' est donc un ensemble tel que, lorsque l'origine de l'élément structurant B est incluse dans A', l'élément B est totalement inclus dans A.

Visuellement le résultat est un rétrécissement de la figure de départ et les figures plus petites que l'élément structurant disparaîtront.

Exemple



*Figure 1.17.* Exemple d'érosion sur une image binaire.

C'est une transformation morphologique qui combine deux ensembles en utilisant la soustraction vectorielle. Si A et B sont deux sous ensembles de E, alors l'érosion de A par B notée AOB est définie par:

 $A\Theta B = \{ c \in E \setminus c + b \in A \forall b \in B \} (1.20)$ 

En pratique, on effectue un ET logique entre les pixels contenus dans la fenêtre utilisée, sauf le pixel central [20].

#### c.2.2.2 Erosion des images à plusieurs niveaux de gris

L'érodé d'un pixel c'est la valeur minimale au voisinage de ce pixel comprise dans l'élément structurant. La définition citée plus haut ne marche que pour les images binaires, mais celle-ci peut s'appliquer également aux images à deux niveaux de gris.

#### Exemple



image érodée



*Figure 1.18*. Exemple de l'érosion sur une image à plusieurs niveaux de gris.

Ceci est réalisé en pratique en promenant sur l'image une fenêtre de taille fixe (l'élément structurant), et en effectuant, pour chaque pixel de l'image un OU logique des pixels formant la fenêtre, à l'exception du pixel central.

Les propriétés de l'érosion sont les suivantes:

$1. A \subseteq B \Rightarrow A \Theta K \subseteq B \Theta K$	(1.21)

- $2. A \supseteq B \Rightarrow D \Theta A \subseteq D \Theta B$  (1.22)
- 3.  $(A \cap B) \ominus K \Rightarrow (A \ominus K) \cap (B \ominus K)$  mais  $(A \cup B) \ominus K \supseteq (A \ominus K) \cup (B \ominus K)$  (1.23)

 $4. A \subseteq B \Theta C \Rightarrow B \supseteq A \bigoplus C$ (1.24)

#### c.3 Ouverture et fermeture

Les opérateurs de dilatation et d'érosion ont tendance à augmenter ou diminuer la taille des objets contenus dans l'image. C'est pour cette raison qu'on a défini deux autres opérateurs qui sont l'ouverture et la fermeture, et qui représentent, une combinaison des opérateurs d'érosion et de dilatation.

- L'ouverture d'une image B, par un élément structurant K, est définie par:

$$B^{\circ} K = (B \Theta K) \bigoplus K.$$
(1.25)

Exemple



Figure 1.19. Exemple d'ouverture sur une image binaire.



image après l'ouverture



Figure 1.20. Exemple d'ouverture sur une image à plusieurs niveaux de gris.

- La fermeture d'une image B, par un élément structurant K, est définie par:

 $B \bullet K = (B \bigoplus K) \Theta K \qquad (1.26)$ 

Exemple



Figure 1.21. Exemple de fermeture sur une image binaire.



*Figure 1.22*. Exemple de fermeture sur une image à plusieurs niveaux de gris.

Les propriétés concernant l'ouverture et la fermeture d'une image sont :

1. $A \oplus K = (A \oplus K) \circ K = (A \bullet K) \oplus K$	(1.27)
2. (A • K) •K = A •K	(1.28)
3. (A ° K) ° K = A ° K	(1.29)
4. A ° B ⊆ A	(1.30)
5. A ⊆ A • B	(1.31)

#### c.4 Autres opérateurs

#### c.4.1 Gradient morphologique (ou Gradient symétrisé)

Le gradient morphologique donne une approximation du module du vecteur gradient en chaque point. Pour une fonction f continûment différentiable [21]:

$$|\nabla f| = \lim_{\lambda \to 0} \frac{(f \oplus K) - (f \Theta K)}{2\lambda}$$
(1.32)

Où k est l'élément structurant associé au disque fermé de rayon  $\lambda$ .

Dans le cadre discret, pour une image A, la norme du gradient s'obtient par la transformation:

$$|\nabla \mathbf{I}| = \frac{(\mathbf{f} \oplus \mathbf{K}) - (\mathbf{f} \Theta \mathbf{K})}{2}$$
(1.33)

Où k est l'élément structurant.
Il suffit donc de dilater et d'éroder l'image en niveaux de gris puisque le gradient s'interprète comme l'étendue, c'est-à-dire la différence entre le « max. » et le « min. » sur le disque unité.

#### c.4.2 Gradient par dilatation

Pour une image A, le module du gradient par dilatation avec l'élément structurant k s'obtient par la transformation:

$$|\nabla^+ A| = (A \bigoplus K) - A \tag{1.34}$$

C'est-à-dire la différence entre le dilaté de l'image et l'image elle même.

### c.4.3 Gradient par érosion

Pour une image A, le module du gradient par érosion avec l'élément structurant k s'obtient par la transformation:

$$|\nabla^{-}A| = A - (A \Theta K)$$
(1.35)

C'est-à-dire la différence entre l'image et son érodée.

#### c.4.4 Laplacien morphologique

Le module du Laplacien  $|\Delta A|$  d'une image A est la différence entre le module du gradient par dilatation et celui du gradient par érosion:

$$|\Delta A| = |\nabla^{+}A| - |\nabla^{-}A|$$
 (1.36)

Nous pouvons aussi utiliser le gradient par érosion ou le gradient par dilatation pour effectuer la détection de contours.



#### Figure 1.23. Opérateurs morphologiques.

#### d Interprétation des résultats

D'après les résultats obtenus, il est clair que l'application des détecteurs de contours du premier et deuxième ordre aux images donne souvent une mauvaise extraction de contours et une sensibilité au bruit.

Le résultat donné par l'application du masque de Prewitt fait apparaître tout de même des points de bruit qui ne font pas partie des contours. C'est un détecteur à grande précision. Le masque de Sobel donne un résultat presque identique au précédent, par contre les contours sont accentués et épais (Figure 1.10).

L'application de l'opérateur MDIF sur l'image donne un résultat proche de celui donné par le masque de Prewitt (Figure 1.11). La différence apparaît dans une sorte de flou obtenu sur l'image résultat. Cela est dû au fait que l'opérateur MDIF est une combinaison d'un filtre moyenneur qui réalise le lissage et d'un détecteur dérivateur utilisant les masques directionnels de Prewitt.

Pour l'opérateur NAGDIF, les contours détectés sont minces et discontinus du fait que les transitions dans l'image originale ont de petites largeurs (Figure 1.13).

Pour les contours obtenus par l'utilisation des gradients morphologiques (Figure 1.23), nous avons globalement une mauvaise extraction de contours et une sensibilité moindre au bruit, mais nous observons l'apparition de quelques points isolés et des contours plus épais.

Le Laplacien morphologique présente une sensibilité au bruit comme tout autre opérateur de dérivation du second ordre.

### e les contours actifs

Les contours actifs tirent leur origine des modèles élastiques [22], mais la communauté s'accorde à les attribuer à l'équipe Kass, Witkin [22] qui introduisirent les snakes ou contour. Les snakes tiennent leur nom de leur aptitude à se déformer comme des serpents. Depuis le travaille de cette équipe, les modèles devenus un sujet très important pour la communauté du traitement d'image.

### e.1 définition

Un contour actif est une courbe qui évolue d'une forme initiale vers les frontières d'un objet d'intérêt, sous l'action d'une force [22]. Donc les contours actifs sont définis par une courbe continue, fermée ou non, à extrémités fixe ou non.

Ils se déforment à partir d'une position d'initialisation située prées de l'objet d'intérêt. Le modèle est soumis à des forces qui le déforment et le déplacent dans l'image. Alors le principe des contours actifs est de faire évaluer un contour initial autour de l'objet d' intérêt vers une position d'équilibre, c'est à dire une détection des bords de l'objet à détecter.

### Exemple



Figure 1.24. Principe des contours actifs.

Le contour actif  $\Gamma$  évolue vers l'objet d'intérêt sous l'action d'une force dirigée s uivant la normale N au contour.

# 1.7 Conclusion

Nous avons présenté dans ce chapitre, des généralités sur le traitement d'image ainsi que les différents blocs qui constituent un système de traitement d'image.

Deux approches ont été abordées concernant le bloc de segmentation: l'approche région et l'approche frontière. Dans cette dernière approche, nous nous somme intéressés à plusieurs méthodes telle que : les méthodes dérivatives, les méthodes morphologiques et la méthode de segmentation par les contours actifs qui fera l'objet du prochain chapitre.

# 2.1 Introduction

Depuis les travaux fondateurs de Kass, Witkin et Terzopoulos [23], les modélisations de type contours et surfaces actifs ont motivé de nombreux développements tant théoriques qu'applicatifs, démontrant leur intérêt pour la résolution de problématiques de reconstruction et de segmentation en imagerie multidimensionnelle.

Les techniques de segmentation d'images basées sur les contours actifs considèrent un objet géométrique (courbe, surface, hyper surface, etc.) qui, à partir d'une forme et d'une position initiale, évolue progressivement pour venir s'appliquer sur les structures à segmenter (figure 2.1). Cette évolution est généralement régie par un certain nombre de contraintes pouvant prendre la forme de champs de vitesses ou de forces inhérentes aux propriétés du modèle et à son environnement. Ces méthodes se basent essentiellement sur des hypothèses relatives à l'intensité de l'image ainsi qu'à la régularité et à la forme des vaisseaux.



*Figure 2.1.* Principe d'un contour actif appliqué à une image coronarographique.

Les contours actifs peuvent être divisés en deux classes selon leur représentation et leur implémentation [21]: Les contours actifs paramétriques et les contours actifs géométriques.

 Les contours actifs paramétriques: Ce sont des courbes v(s)=[x(s), y(s)], où s est l'abscisse curviligne, qui peuvent se déformer progressivement de manière à s'approcher des contours d'un objet. Cette déformation est guidée par la minimisation d'une fonctionnelle d'énergie comprenant deux termes:

- Une énergie interne  $E_{int}\, qui \, permet \, de \, régulariser le contour.$ 

- Une énergie externe  $E_{ext}$  reliée à l'image et aux contraintes particulières que l'on peut ajouter.

2. Les contours actifs géométriques: Ce sont des modèles basés sur la théorie d'évolution de courbes « Curve Evolution Theory » et la méthode d'ensemble de niveaux « Level Set Method ». Dans cette classe de contours, chaque point de la courbe se déplace dans la direction de son vecteur normal à une vitesse proportionnelle à la courbure. Cette approche est une alternative à la minimisation d'énergie qui peut être vue comme la recherche de la solution d'une équation de Hamilton-Jacobi, et être ainsi résolue de façon efficace par l'utilisation de la méthode des ensembles des niveaux « Level Set » d'Osher et de Sethian [24].

### 2.2 Les contours actifs paramétriques

Comme il a été décrit précédemment, ces modèles sont construits à partir d'une fonction de paramètres ayant des informations sur la forme recherchée. Ils possèdent une structure supplémentaire ajoutant ainsi d'autres contraintes pour contrôler l'évolution dans l'image. Ils sont développés pour des applications souvent très particulières [22].

### 2.2.1 Les contours actifs traditionnels

Les contours actifs traditionnels, proposés par Kass, sont des courbes  $v(s) = [x(s),(y(s)], s \in [0, 1]$  qui se déplacent à travers un domaine spatial de l'image pour minimiser l'énergie donnée par :

$$E = \int_0^1 \left[ \frac{1}{2} \left( \alpha \left| v'(s) \right|^2 + \beta \left| v''(s) \right|^2 \right) + E_{ext}(v(s)) \right] ds$$
 (2.1)

La première partie de l'intégrale représente l'énergie interne du Snake  $E_{int}$ , dont la dérivée du premier ordre v'(s) contrôle l'étirement du Snake et la deuxième v''(s) contrôle la flexion,  $\alpha$  et  $\beta$  sont les poids qui contrôlent respectivement la tension et la rigidité.

La deuxième partie de l'intégrale représente l'énergie externe du snake  $E_{ext}$ , elle correspond à l'adéquation aux données. Cette énergie prend en compte les caractéristiques de l'image, l'énergie externe peut être donnée par la relation suivante:

$$E_{\text{ext}}(x, y) = - |\nabla [G(x, y, \sigma) * I(x, y)]|^2$$
(2.2)

Où G(x, y,  $\sigma$ ) est la fonction Gaussienne à deux dimensions d'écart type  $\sigma$  [18]:

G(x, y, 
$$\sigma$$
) =  $-\frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2}\left[\frac{x^2+y^2}{\sigma^2}\right]\right\}$  (2.3)

'\*' Est l'opérateur de convolution et '∇' est l'opérateur gradient.

On peut trouver la valeur qui minimise l'équation (2.1) en résolvant l'équation d'Euler-Lagrange en v :

$$\alpha v^{(2)}(s) - \beta v^{(4)}(s) = \nabla E_{ext}$$
 (2.4)

Cette équation peut être représentée comme une équation de force d'équilibre, avec  $[\alpha v^{(2)}(s) - \beta \ v^{(4)}(s)] \text{ correspondant aux forces internes } F_{int} \text{ et } (-\nabla \ E_{int} \ ) \text{ aux forces}$ externes  $F_{ext}^{(p)}$  ce qui donne :

$$F_{int} - F_{ext}^{(p)} = 0$$
 (2.5)

 $F_{int}$  maintient la cohésion des points et la raideur du snake, alors que  $F_{ext}^{(p)}$  tire le snake aux bords souhaités de l'image.

Le comportement du snake peut être modélisé par une fonction qui dépend du temps V(s, t) ce qui donne la solution de l'équation (2.5) :

$$\gamma \frac{\partial v(s,t)}{\partial t} = \alpha v^{(2)}(s,t) - \beta v^{(4)}(s,t) - \nabla E_{ext}$$
(2.6)

Ou  $v^{(2)}$  et  $v^{(4)}$  sont respectivement la deuxième et la quatrième dérivée de v par rapport à s, le paramètre  $\gamma$  représente la viscosité de la courbe, si  $\gamma$  est grand, le comportement du snake devient lent.

Pour parvenir à une solution de l'équation (2.6), le terme  $\frac{\partial v(s,t)}{\partial t}$  doit être égal à zéro, donc:

$$||F_{int}|| = ||F_{ext}^{(p)}||$$
 (2.7)

La version discrète de l'équation (2.6) est obtenue en discriminant la courbe paramétrique à chaque instant t :  $V_t = \{ V_{i,t} = (x_{i,t}, y_{i,t}), i=1...n \}$ , où n représente le nombre de points dans le contour, nous pouvons approximer la première dérivée par :

$$v_{i,t} = v_{i,t} - v_{i-1,t}$$
 (2.8)

Et la deuxième dérivée par :

$$v''_{i,t} = v_{i+1,t} - 2 v_{i,t} + v_{i-1,t}$$
 (2.9)

La dérivée temporelle est approximée par :

$$\frac{\partial v_{i,t}}{\partial t} = v_{i,t} - v_{i,t-1}$$
(2.10)

Donc la solution peut être écrite sous la forme vectorielle suivante :

$$\gamma \frac{\partial V_t}{\partial t} - \alpha V_t^{(2)} + \beta V_t^{(4)} = -\frac{\partial E_{ext}(V_{t-1})}{\partial v}$$
(2.11)

Pour chaque point  $v_{i,t}$ , en prenant  $\alpha$  et  $\beta$  constants, on obtient:

$$\gamma (v_{i,t} - v_{i,t-1}) - \alpha (v_{i+1,t} - 2v_{i,t} + v_{i-1,t}) + \beta (v_{i+2,t} - 4v_{i+1,t} + 6v_{i,t} - 4v_{i-1,t} + v_{i+2,t})$$
$$= -\frac{\partial E_{ext}(V_{t-1})}{\partial v}$$
(2.12)

Cette équation est la plus utilisée pour une éventuelle implémentation et sa solution peut être obtenue à chaque itération.

### 2.2.2 Conclusion

Les contours actifs paramétriques (traditionnels) présentent trois inconvénients majeurs:

- En premier lieu, les contours actifs paramétriques dépendent de l'initialisation. Si le contour est initialisé loin de l'objet souhaité, il a peu de chance, de rejoindre le contour désiré. De plus, si l'image est bruitée, il a de fortes chances de s'arrêter sur des contours parasites.

- Le réglage des coefficients n'est pas pris en charge théoriquement et est généralement effectué de façon heuristique. Les paramètres d'élasticité et de courbure adaptés à une forme particulière ne sont donc pas utilisables sur une autre forme. On peut remarquer qu'une forte valeur de  $\alpha$  induit des contraintes de tension importantes. De la même manière, une forte valeur de  $\beta$  efface les détails du contour. Il est donc nécessaire de procéder à des essais préalables pour déterminer le meilleur jeu de paramètres, ce qui est handicapant dans le cas d'une segmentation automatique.

- De plus, ce type de contours a des difficultés de gérer automatiquement le changement de topologie de la courbe en cours d'évolution.

Nous allons dans la partie suivante développer deux techniques appartenant aux contours actifs géométriques, afin de résoudre les problèmes rencontrés dans le cas des contours actifs paramétriques.

## 2.3 Les contours actifs Géométriques

### 2.3.1 Introduction

Comme nous l'avons vu dans le paragraphe précédent, les contours actifs paramétriques présentent quelques inconvénients, par exemple, leur sensibilité à la position et à la forme du contour initial, ainsi que leur difficulté de gérer automatiquement le changement de topologie de la courbe en cours d'évolution. Pour résoudre les problèmes cités dans le paragraphe précédent, Caselles et al [22]. ont introduit un modèle géométrique des contours actifs formulé par des équations aux dérivées partielles (EDPs) d'évolution de courbes. Chaque point de la courbe se déplace dans la direction de son vecteur normal avec une vitesse proportionnelle à la courbure. Cette approche est une alternative à la minimisation d'énergie qui peut être vue comme la recherche de la solution d'une équation de Hamilton-Jacobi, et être ainsi résolue de façon efficace par l'utilisation de la méthode des courbes de niveaux « Level Set » d'Osher et de Sethian. La formulation des contours actifs par les ensembles de niveaux[24](Level Sets) permet d'implémenter les contours actifs tout en gérant le problème des changements de topologie, de plus, son extension aux dimensions supérieures est aisée. De ce fait, l'application de la formulation par level-set à la segmentation des images est devenue extrêmement populaire à cause de sa capacité à capturer la topologie des formes et à identifier les caractéristiques spécifiques dans les images.

Nous allons présenter dans cette partie deux méthodes appartenant à la classe géométrique, qui sont respectivement : Level-sets traditionnels (avec réinitialisation) et level-sets sans réinitialisation de la fonction de distance. Nous procéderons par la suite à une étude comparative permettant d'évaluer les avantages et les inconvénients de chaque méthode à partir des résultats obtenus.

### 2.3.2 Méthode d'ensemble de niveau zéro (Level sets traditionnels)

La méthode des ensembles de niveau zéro est une méthode de simulation numérique utilisée pour l'évolution des courbes et des surfaces dans les domaines discrets. Cette méthode permet de faire évoluer une courbe paramétrique fermée C(t) suivant une équation appelée « équation d'ensemble de niveaux ».

Chaque point de la courbe C évolue suivant la direction normale à la courbe avec une vitesse F (figure 2.2).



Figure 2.2. Evolution de la courbe C.

Pour une surface 2D, la courbe C(t) peut être représentée comme un niveau d'ensemble d'une fonction à trois dimensions (hyper surface) { $\phi$  (x, y, t)=0}. L'objectif est de produire une fonction d'évolution  $\phi$ (x, y, t) qui contient le mouvement embarqué de  $\phi$ (t) quand le niveau d'ensemble { $\phi$  =0}.

Soit  $\phi(x, y, t=0)$  définie comme suit:

$$\phi(x, y, t=0) = \pm d$$
 (2.13)

Avec d est la fonction de distance du point (x, y) par rapport à l'interface C(t=0), et le signe plus (moins) est choisi lorsque le point (x, y) est à l'extérieur (intérieur) de l'objet d'intérêt.

L'avantage principal de cette méthode est la possibilité de gérer automatiquement le changement de topologie en cours d'évolution de la courbe, cette dernière peut être divisée en deux ou trois courbes, inversement plusieurs courbes peuvent fusionner et devenir une seule courbe. La méthode des ensembles de niveau zéro est capable de réaliser ce changement en utilisant une fonction d'ordre supérieur :  $R^{n+1} \rightarrow R^n$ pour simuler C, et donc, le niveau zéro (C) peut se diviser pendant que la fonction  $\phi$ reste continue.



*Figure 2.3.* Changement de topologie au cours De l'évolution de la courbe.

La fonction d'ordre supérieur utilisée est la distance algébrique à la courbe C. Alors, si notre courbe C est paramétrique plane (dimension 1) la fonction utilisée pour sa représentation et pour simuler son évolution sera  $\phi$  (x, y) (dimension 2) à valeurs réelles.

Pour une courbe plane C, un tableau bidimensionnel de nombres réels  $\phi$  (x, y) est utilisé, l'évolution de la courbe signifie la mise à jour de tout le tableau  $\phi$  (x, y).

La fonction  $\phi(x, y)$  évolue selon l'équation d'ensemble de niveaux suivante :

$$\frac{\partial \Phi}{\partial t} = F |\nabla \Phi| \qquad (2.14)$$

Où : F est une fonction scalaire de vitesse dépendant des facteurs suivants :

- Les propriétés locales de la courbe.
- Les paramètres externes liés au gradient d'image.
- Les termes additionnels de propagation de la courbe.

L'ensemble de ces facteurs peut être écrit comme suit :

$$F = F_{\text{prop}} + F_{\text{curv}} + F_{\text{adv}}$$
(2.15)

- $F_{prop} = F_o$  est la vitesse de propagation de la courbe.
- $F_{curv} = -\varepsilon K$  est la vitesse liée aux propriétés locales de la courbe.
- $F_{adv} = U(x, y, t)$  . n est la vitesse d'advection.

Avec n est la normale unitaire à la courbe donnée par :  $\vec{n} = \frac{\nabla \varphi}{|\nabla \varphi|}$ .

L'équation d'ensemble de niveaux peut être écrite comme suit :

$$\frac{\partial \phi}{\partial t} + F_0 |\nabla \phi| + U(x, y, t). \nabla \phi = \varepsilon k |\nabla \phi| \qquad (2.16)$$

- Le terme  $F_0 |\nabla \phi|$  dans l'équation décrit le mouvement des levels sets dans la direction normale à la courbe, il dépend du terme g ( $|\nabla \phi|$ ), qui est une fonction monotone décroissante de l'amplitude du gradient de l'image I. Ce terme peut être écrit selon l'équation suivante:

g (
$$|\nabla \varphi|$$
) =  $\frac{1}{1 + |\nabla G_{\sigma^*} I|^p}$  (2.17)

Où I(x, y)est l'image, et G est un noyau Gaussien de variance  $\sigma$ .

'\*' : est l'opération de convolution, avec p=1 ou p=2.

Le terme  $F_0 |\nabla \varphi|$  est défini par :

$$F_0 |\nabla \varphi| = \alpha [\max (F_{0ij}, 0) \nabla^+ + \min (F_{0ij}, 0) \nabla^-]$$
 (2.18)

Le coefficient  $\alpha$  peut être positif ou négatif, en fonction de la position relative entre le contour initial et l'objet d'intérêt. Par exemple, si le contour initial est placé en dehors de l'objet d'intérêt, le coefficient de pondération  $\alpha$  devrait prendre une valeur positive, de sorte que le contour puisse se rétrécir plus rapidement. Si le contour initial est placé à l'intérieur des contours de l'objet d'intérêt, le coefficient de pondération  $\alpha$  devrait prendre une valeur négative pour accélérer l'expansion du contour.

Où  $F_{0ij}$  dépend du terme g ( $|\nabla \varphi|$ ) et  $\nabla^+$  et  $\nabla^-$  sont définis par :

$$\nabla^{+} = [\max(D_{ij}^{-x}\phi, 0)^{2} + \max(D_{ij}^{+x}\phi, 0)^{2} + \min(D_{ij}^{-y}\phi, 0)^{2} + \min(D_{ij}^{+y}\phi, 0)^{2}] \quad (2.19)$$

$$\nabla^{-} = [\max(D_{ij}^{+x}\phi, 0)^{2} + \min(D_{ij}^{-x}\phi, 0)^{2} + \max(D_{ij}^{+y}\phi, 0)^{2} + \min(D_{ij}^{-y}\phi, 0)^{2}] \quad (2.20)$$

L'implémentation numérique de cette évolution fait appel aux approximations des dérivées par la méthode des différences finies du premier ordre (pour notre cas en 2D),  $D_{ij}^{+x}$ ,  $D_{ij}^{+y}$ , sont les FD "Forward Differences" de  $\phi$  selon x et y, elles sont définies comme suit :

$$D_{ij}^{+x} = D_{ij}^{+x} \Phi = \frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x}$$
 (2.21.a)

$$D_{ij}^{+y} = D_{ij}^{+y} \Phi = \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y}$$
 (2.21.b)

 $D_{ij}^{-x}$ ,  $D_{ij}^{-y}$ , sont les BD "Backward Differences" de  $\phi$  selon x et y, elles sont définies comme suit :

$$D_{ij}^{-x} = D_{ij}^{-x} \phi = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}$$
 (2.22.a)

$$D_{ij}^{-y} = D_{ij}^{-y} \Phi = \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y}$$
 (2.22.b)

- Le terme  $F_{adv}$   $|\nabla \varphi|$  correspond à la vitesse d'advection, il peut être exprimé sous la forme suivante :

 $F_{adv} |\nabla \varphi| = [\max(u_{ij}^{n}, 0) D_{ij}^{-x} + \min(u_{ij}^{n}, 0) D_{ij}^{+x} + \max(v_{ij}^{n}, 0) D_{ij}^{-y} + \min(v_{ij}^{n}, 0) D_{ij}^{+y}] (2.23)$   $\rightarrow$ Où U = (u, v) avec u, v sont respectivement les dérivées spatiales de g ( $|\nabla \varphi|$ ).

 $D_{ij}^{+x},\,D_{ij}^{+y},\,D_{ij}^{-x},\,D_{ij}^{-y}\,$  sont respectivement les FD et BD de  $\varphi\,$  selon les directions de x et y.

- Le terme  $F_{curv} |\nabla \phi|$  dépend de la courbure K, pour l'implémentation des contours actifs géométriques, le calcul de la courbure locale en chaque point de la courbe est nécessaire.

Cependant, ce calcul est coûteux en temps. De plus, pour certaines implémentations d'évolution des courbes, il est nécessaire de calculer la courbure non seulement sur la courbe mais dans tout le domaine d'évolution. Cela est très coûteux, même avec les méthodes à bandes étroites[22] qui ne mettent à jour qu'une partie du domaine délimité par deux courbes (les bandes) qui entourent la courbe en évolution. La courbure locale dans un point de C peut être calculée à partir de  $\phi$  en utilisant les différences finies centrales avec la formule [21]:

$$k = \nabla \left(\frac{\nabla \phi}{|\nabla \phi|}\right) = \frac{\phi_{xx} \phi_y^2 - 2\phi_{xy} \phi_x \phi_y + \phi_{yy} \phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}$$
(2.24)

Et le terme  $F_{curv}$   $|\nabla \phi|$  devient :

$$F_{\text{curv}} \left| \nabla \phi \right| = -\varepsilon \left[ \frac{\phi_{xx} \phi_y^2 - 2\phi_{xy} \phi_x \phi_y + \phi_{yy} \phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \cdot ((D_{ij}^{0x})^2 + (D_{ij}^{0y})^2)^{1/2} \right]$$
(2.25)

Où  $D_{ij}^{0x}$  et  $D_{ij}^{0y}$  sont les CD "Centered Differences", elles sont définies comme suit :

$$D_{ij}^{0x} = D_{ij}^{0x} \Phi = \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2\Delta x}$$
(2.26.a)

$$D_{ij}^{0y} = D_{ij}^{0y} \Phi = \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2\Delta y}$$
(2.26.b)

La dérivée temporelle est donnée par l'équation suivante :

$$\frac{\partial \phi}{\partial t} = \frac{\phi_{ij}^{k+1} - \phi_{ij}^{k}}{\tau}$$
(2.27)

Avec  $\tau$  est le pas relatif au temps.

En remplaçant les trois termes dans cette l'équation, on obtient l'équation suivante [13]:

$$\begin{split} &\varphi_{ij}^{n+1} = \varphi_{ij}^{n} + \left[-\alpha \left[\max\left(F_{0ij}, 0\right)\nabla^{+} + \min\left(F_{0ij}, 0\right)\nabla^{-}\right] - \left[\max\left(u_{ij}^{n}, 0\right)D_{ij}^{-x} + \min\left(u_{ij}^{n}, 0\right)D_{ij}^{+x}\right] + \max\left(v_{ij}^{n}, 0\right)D_{ij}^{-y} + \min\left(v_{ij}^{n}, 0\right)D_{ij}^{+y}\right] + \varepsilon\left[K^{*}\left((D_{ij}^{0x})^{2} + (D_{ij}^{0y})^{2}\right)^{1/2}\right] \end{split}$$
(2.28)

### a Construction de la fonction de distance

La fonction d'ensemble de niveaux  $\phi$  doit satisfaire la relation:  $|\nabla \phi| = 1$ , en préservant sa configuration comme une fonction de distance signée de front C(t), durant tout le temps de déformation [21]. Ainsi, si la configuration initiale de C(t) est un point simple,  $\phi$  est construite pour s'assurer que tous ses ensembles de niveaux sont équidistants à l'ensemble de niveau zéro de C(t). Supposons que la fonction de vitesse est une constante uniforme qui prend la valeur 1 agissant sur la fonction

d'ensemble de niveaux qui est définie par  $\phi$ , nous pourrions prévoir que le point initial augmenterait indéfiniment à chaque itération. Cette première étape est l'idée fondamentale de la méthode d'ensemble de niveau zéro. La deuxième étape consiste à utiliser la fonction de vitesse, comme le montre l'équation (2.14) pour faire évoluer  $\phi$  vers les contours désirés.

Cette complexité est brièvement discutée dans les références suivantes : Une des solutions présentées doit périodiquement arrêter les calculs et reconstruire la fonction de distance signée  $\phi$  selon la position actuelle de C(t). Le procédé est complexe et coûteux, car il devrait répéter chaque itération pour éviter des erreurs. Une deuxième solution propose d'évoluer une fonction de vitesse prolongée  $F_{ext}$  égale à la fonction scalaire F ainsi que la fonction  $\phi$ . Puisque  $F_{ext}$  devrait également correspondre à une fonction de distance signée, elle devrait satisfaire la condition suivante :

$$F_{\text{ext}} \nabla \phi = 0 \tag{2.29}$$

Nous pouvons, ainsi, utiliser la méthode Fast Marching [18] pour calculer la fonction de distance signée  $\phi$  en résolvant l'équation d'Eikonal suivante :

$$|\nabla \mathsf{T}| = 1 \tag{2.30}$$

Pour chaque partie de l'interface, la condition de frontière (T = 0) doit être vérifiée sur l'ensemble de niveau zéro de la fonction  $\phi$ . La solution T sera ensuite notre fonction de distance signée  $\phi^{temp}$  temporaire.

#### *b Réinitialisation de la fonction de distance:*

Il est possible de réinitialiser périodiquement  $\phi$  comme une fonction de distance signée tout en satisfaisant la relation:  $|\nabla \phi| = 1$ . Ceci est obtenu en résolvant l'équation suivante :

$$\frac{d\Phi}{dt} = \operatorname{sign}(\Phi_0) \left(1 - |\nabla \Phi|\right)$$
(2.31)

Où  $\phi_0$  est la fonction à réinitialiser, sign $(\phi_0)$  est le signe de la fonction.

En tenant compte des valeurs initiales de  $\phi$ , la résolution de l'équation (2.31) à l'état d'équilibre donne de nouvelles valeurs de  $\phi$  avec la condition  $|\nabla \phi| = 1$ , et donc le terme sign $(\phi_0)$  (1- $|\nabla \phi|$ ) égal à 0.

L'inconvénient principal de cette approche est le temps de calcul pour maintenir la stabilité de la courbe durant l'évolution et assurer les résultats désirés. Sussman et Fatemi [25], ont démontré que le problème de cette approche est lié au nombre d'itérations plus grand, et de plus le niveau zéro de la fonction de distance peut s'éloigner de la position prévue. Ainsi, lorsque la fonction d'ensemble de niveaux est différente de celle de distance signée [27], les méthodes de réinitialisation peuvent ne pas être en mesure de réinitialiser la fonction d'ensemble de niveaux à une fonction de distance signée. En pratique l'évolution de la fonction d'ensemble de niveaux à une sixeaux peut s'écarter considérablement de sa valeur en tant que distance signée durant un petit nombre d'itérations, surtout lorsque le pas de temps choisi n'est pas assez petit [26]. Du point de vue pratique, le processus de réinitialisation peut être assez compliqué et coûteux à cause de la complexité des approximations numériques de cette approche.

Pour éviter les problèmes ci-dessus, Li Chunming [27] a proposé une nouvelle formulation variationnelle qui peut être facilement mise en œuvre par de simples différences finies et élimine définitivement par conséquent le besoin de réinitialisation.

### 2.3.4 Formulation variationnelle des levels sets sans la

#### réinitialisation de la fonction de distance

Pour maintenir la stabilité de la courbe durant l'évolution, Chunming [27] a proposé l'intégrale suivante :

$$p(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dx dy$$
 (2.32)

Comme nous avons montré précédemment, la fonction de distance doit satisfaire la condition  $|\nabla \phi| = 1$ , de ce fait, l'intégrale  $p(\phi)$  évalue la métrique pour déterminer le

degré de convergence de la fonction  $\phi$  vers la fonction de distance signée dans un domaine  $\Omega \subset R^2$ .

La nouvelle formulation variationnelle peut être écrite comme suit :

$$\zeta(\mathbf{\phi}) = \mu \,\mathsf{p}(\mathbf{\phi}) + \xi_{q,\,\lambda,\,\nu}(\mathbf{\phi}) \tag{2.33}$$

Avec  $\mu > 0$  et représente le paramètre qui contrôle l'effet d'écart de  $\phi$  par rapport à la fonction de distance signée.

L'énergie  $p(\phi)$  est l'énergie interne de  $\phi$ , elle dépend uniquement de  $\phi$ , tandis que l'énergie  $\xi_{g, \lambda, v}(\phi)$  dépend des données de l'image (comme la vitesse de propagation  $F_{prop}$  dans le cas des levels sets traditionnels), cette énergie pousse l'ensemble de niveau zéro vers les bords des objets, elle peut être définie comme suit :

$$\xi_{g,\lambda,\nu}(\phi) = \lambda L_g(\phi) + \nu A_g(\phi)$$
(2.34)

Avec g, l'image contour définie dans l'équation (2.17) et  $\lambda > 0$  et v sont des constantes. Les termes  $L_q(\phi)$  et  $A_q(\phi)$  sont définies selon les équations (2.35) et (2.36) :

$$L_{g}(\phi) = \int g \,\delta_{\varepsilon}(\phi) \qquad \qquad |\nabla \phi | \,dx \,dy \qquad (2.35)$$

$$A_g(\phi) = \int g H(-\phi) \qquad dx \ dy \qquad (2.36)$$

La fonction d'énergie  $L_g(\phi)$  calcule la longueur de la courbe de niveau zéro de la fonction  $\phi$  selon la métrique euclidienne ds = |C'(q)|dq, où C(q): [0, 1]  $\rightarrow R^2$  est une courbe plane paramétrée.

La solution de  $L_g(\phi)$  évalue le mouvement de chaque point du contour actif, afin de diminuer la longueur deL<sub>g</sub>.

La fonction d'énergie  $A_g(\phi)$  dans l'équation (2.36) est introduite pour accélérer l'évolution de la courbe. Le coefficient v de  $A_g(\phi)$  dépend de la position relative du contour initial par rapport à l'objet désiré. Par exemple si le contour initial est placé en dehors de l'objet d'intérêt, le coefficient v devrait prendre une valeur positive, de sorte que le contour puisse se rétrécir plus rapidement. Et si le contour initial est placé à l'intérieur des contours de l'objet d'intérêt, le coefficient v devrait prendre une valeur négative pour accélérer l'expansion du contour.

La fonction de dirac  $\delta_\epsilon est$  définie comme suit :

$$\delta_{\varepsilon}(\mathbf{x}) = \begin{cases} 0 & |\mathbf{x}| > \varepsilon & (2.37) \\ \\ \frac{1}{2\varepsilon} \left[ 1 + \cos\left(\frac{\pi \mathbf{x}}{\varepsilon}\right) \right] & |\mathbf{x}| \le \varepsilon \end{cases}$$

Le gradient qui minimise la fonction (2.33) est donné par l'équation suivante :

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \zeta}{\partial \phi} = \frac{\partial (\mu p(\phi) + \xi_{g, \lambda, v}(\phi))}{\partial \phi}$$
(2.38)

En utilisant les calculs variationnels, la première dérivée de la fonction  $\zeta$  peut être écrite sous la forme suivante:

$$\frac{\partial \zeta}{\partial \phi} = -\mu \left[ \Delta \phi - \operatorname{div}(\frac{\nabla \phi}{|\nabla \phi|}) \right] - \lambda \delta_{\varepsilon}(\phi) \operatorname{div}(g \frac{\nabla \phi}{|\nabla \phi|}) - \nu g \delta_{\varepsilon}(\phi) \quad (2.39)$$

Δ : est l'opérateur Laplacien.

Notons que la dérivée temporelle  $\frac{\partial \Phi}{\partial t}$  est approximée par les « Forward Differences » FD.

Pour la minimisation de la fonction d'énergie  $\zeta$ , l'équation d'Eurler-Lagrange doit être vérifiée.

En utilisant la dérivée temporelle  $\frac{\partial \Phi}{\partial t}$ , l'équation d'évolution (2.39) peut être réécrite sous la forme suivante :

$$\frac{\phi_{ij}^{k+1} - \phi_{ij}^{k}}{\tau} = \mu \left[ \Delta \phi - \text{div}(\frac{\nabla \phi}{|\nabla \phi|}) \right] + \lambda \delta_{\varepsilon}(\phi) \, \text{div}(g \frac{\nabla \phi}{|\nabla \phi|}) + v g \, \delta_{\varepsilon}(\phi) \qquad (2.40)$$

 $\tau$  : est le pas de temps.

Le terme  $\mu [\Delta \phi - \text{div} (\frac{\nabla \phi}{|\nabla \phi|})]$  de l'équation (2.40) correspond au gradient de la fonction d'énergie interne  $\mu p(\phi)$ , l'effet de ce terme est de maintenir la stabilité de  $|\nabla \phi|=1$  durant toute l'évolution de la fonction d'ensemble de niveaux.

D'autre part, le terme  $\lambda \, \delta_{\varepsilon}(\phi) \operatorname{div}(g \frac{\nabla \phi}{|\nabla \phi|}) + v g \, \delta_{\varepsilon}(\phi)$  de l'équation (2.40) correspond au gradient des fonctions d'énergie  $\lambda L_g(\phi)$  et  $v A_g(\phi)$ .

#### a Initialisation flexible de la fonction d'ensemble de niveaux

Contrairement à la méthode d'ensemble de niveaux traditionnelle, dont l'initialisation de la fonction  $\phi$  comme une fonction de distance signée est nécessaire, la méthode d'ensemble de niveaux sans réinitialisation n'a pas besoin d'initialiser la fonction  $\phi$  comme une fonction de distance signée. Pour cela, Chunming [27] a proposé la formule suivante pour initialiser la fonction  $\phi_0$ :

$$\Phi_{0}(\mathbf{x}, \mathbf{y}) = \begin{cases} -\rho & (\mathbf{x}, \mathbf{y}) \in \Omega_{0} - \partial \Omega_{0} & (2.41) \\ 0 & (\mathbf{x}, \mathbf{y}) \in \partial \Omega_{0} \\ \rho & \Omega - \Omega_{0} \end{cases}$$

Où  $\Omega$  est le domaine de l'image I.

 $\Omega_0~$  est un sous-domaine de l'image I.

 $\partial \Omega_0$  représente les points de contours de  $\Omega_0$ .

 $\rho$  >0 est une constant.

Cette fonction initiale d'ensemble de niveaux  $\phi_0$  est calculée à partir d'une région arbitraire  $\Omega_0$  dans un domaine de l'image  $\Omega$ .

## 2.4 Conclusion:

La segmentation des images par ensembles de niveaux (Level-Set Method) est devenue extrêmement populaire à cause de sa capacité à capturer la topologie des formes et à identifier les caractéristiques spécifiques dans les images. Contrairement à la méthode d'ensemble de niveaux traditionnelle, dont l'initialisation de la fonction  $\phi$ comme une fonction de distance signée est nécessaire, la méthode d'ensemble de niveaux sans réinitialisation n'a pas besoin d'initialiser la fonction  $\phi$  comme une fonction de distance signée.

Le processus de réinitialisation peut être assez compliqué et coûteux à cause de la complexité des approximations numériques de cette approche.

Dans le chapitre suivant nous allons appliquer l'algorithme de segmentation des images par ensemble des niveaux sans réinitialisation sur des images synthétiques et des images réelles afin de montrer la robustesse cet algorithme de segmentation en le simulant avec le logiciel MATLAB.

# **Chapitre3** Simulation de l'application sous

# Matlab

# 3.1 Introduction

On ne peut conclure sur l'efficacité ou non d'un algorithme que par sa mise à l'épreuve. Apres avoir décrit l'approche théorique de l'algorithme de segmentation des images par ensemble des niveaux sans réinitialisation cité au chapitre 2, vient l'étape d'expérimentation.

Dans ce chapitre, nous allons utiliser des différentes images tests (Synthétique, IRM et image réelle) pour montrer la robustesse de l'algorithme choisi de segmentation des images par ensemble des niveaux sans réinitialisation. Nous avons simulé cet algorithme avec le logiciel MATLAB.

# 3.2 L'outil de développement

MATLAB est un langage de calcul scientifique de haut niveau et représente un environnement interactif pour le développement d'algorithmes, la visualisation et l'analyse de données, ou encore le calcul numérique (MATLAB veut dire *MATrix LABoratory*). En utilisant MATLAB, nous pouvons résoudre des problèmes de calcul scientifique plus rapidement qu'avec les langages de programmation traditionnels.

Le langage MATLAB met à la disposition du développeur les opérations vectorielles et matricielles, fondamentales pour les problèmes d'ingénierie et scientifiques. Il permet un développement et une exécution rapide.

Avec le langage MATLAB, nous pouvons programmer et tester des algorithmes plus rapidement qu'avec les langages traditionnels, car il n'est pas nécessaire d'effectuer les tâches de programmation de bas niveau, comme la déclaration des variables, la spécification des types de données et l'allocation de la mémoire.

# 3.3 Description de l'algorithme de contour actif géométrique

# sans réinitialisation de la fonction de distance

1. Initialisation : On fait l'initialisation flexible de  $\phi$  manuellement à l'aide de la souris

2. Boucle :

Calcul des énergies (interne et externe) : (p ( $\phi$ ) ,  $\xi_{g, \lambda, v}$  ( $\phi$ ) ).

Évolution de la fonction de distance : en utilisant l'équation (2.40).

3. Convergence : (Nécessite une condition d'arrêt).

Arrêt de l'algorithme lorsque la courbe C(t) s'applique sur les objets désirés.

# 3.4 Méthodologie de travail

On commencera notre travaille par une organisation d'ensemble image testé suivie d'un prétraitement, avant d'appliqué l'algorithme de l'ensemble des niveaux. Le premier pas de cet algorithme est la création du contour initial, ensuite on calcule les différentes énergies (interne et externe), à la fin nous appliquons l'équation d'évolution de la courbe, la convergence de cette dernière dépend de la fonction d'arrêt pour extraire l'objet a segmenté.



Figure 3.1. Chaine de traitement pour la segmentation.

L'application d'algorithme et nos tests ont été réalisés sous le système d'exploitation Windows 7, sur un PC doté d'un processeur Intel Core i5 avec 6,00 Go.

# 3.4.1 Ensemble d'image testé

Au début, nous avons enregistré les images dans un fichier qui contient touts les programmes (initialisation, les énergies, etc.) pour éviter les erreurs d'emplacement de nos images choisies.



Figure 3.2. Quelques images de la base de données.

# 3.4.2 Prétraitement

Cette étape peut être appliquée différemment selon les images et leurs bruits, dans le cas d'une image IRM on fait un filtrage anisotropique pour le lissage mais si nous utilisons un autre type d'image la nature de filtrage ou bien la technique de prétraitement changera.

# 3.4.3 Initialisation

L'initialisation c'est une étape primordiale pour le démarrage du processus de segmentation, nous allons utiliser une fonction de l'outil MATLAB qui permettra d'obtenir une courbe initiale, cette méthode se fait en premier lieu par sélection manuelle de quelque points contour initiale. Ensuite toutes les coordonnées des points seront récupérées dans une structure de données représentant notre contour initial.

IL faut bien choisir la position de la courbe initiale soit à l'intérieur de l'objet d'intérêt ou l'inverse, pour cela il ya des paramètres signés qui permettent de donner le sens d'évolution.

### a Résultats d'initialisation :



*Figure 3.3.* Initialisations du contour.

1. Si le contour initial est placé à l'extérieur ou en dehors de l'objet d'intérêt exemple (image.1, image .3), le coefficient  $\nu$  de l'énergie de l'équation (2.34), devrait prendre une valeur positive, de telle sorte que le contour puisse se rétrécir plus rapidement.

2. Si le contour initial est placé à l'intérieur de l'objet d'intérêt exemple (image.2, image.4), le coefficient  $\nu$  devrait prendre une valeur négative pour accélérer l'expansion du contour.

# 3.4.4 Calcule des énergies

Avant l'application de l'équation d'évolution (2.40), il faut calculer toutes les termes qui forment cette équation tell que :

1. Les deux énergies, l'énergie interne et l'énergie externe, qui gèrent l'évolution de la courbe initiale, dans notre travaille nous allons calculer la dérivée de ces deux derniers par rapport à la courbe initiale  $\phi$  de l'équation (2.13) afin de connaitre l'évolution de la courbe.

2. Nous calculons trois fonctions : la courbure, la fonction d'arrêt g (2.17) et la fonction de dirac de  $\phi$  de l'équation (2.35).

# 3.4.5 Evolution de la courbe initiale

La méthode du contour actif consiste à faire évoluer un contour initial en minimisant la fonctionnelle d'énergie, c'est-à-dire que 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 point voisin dont la fonctionnelle d'énergie est la plus faible. Ce qui revient à

chercher un ensemble de point pour lequel l'énergie est minimale. Donc on déplace un point unique pour constituer un nouveau contour actif pour chaque itération.

La déformation du contour actif est faite selon l'algorithme suivant:

Début

•	Faire	
•		Pour chaque point du contour initial
•		. Pour tous les points du voisinage
•		Calculer les énergies
•		. Fin pour
	•	. Minimiser pour obtenir le nouveau point
	•	Fin pour
	Jusqu'	à critère d'arrêt
Fin		

Nous avons considéré les paramètres suivants :

 $\Delta t = 1/7$ ,  $\sigma = 1.5$ , v = -1.5,  $\gamma = 3$ ,  $\mu = 0.2/5$ , avec  $\rho = 4$ .

# 3.5 Application d'algorithme sur les déférents types d'image

La présente partie sera consacré à une série de tests sur les différentes types d'images tests (synthétique, IRM et réelle), qui seront réalisé dans l'ordre suivant :

1. Test sur la détection du contour par level set avec initialisation en dehors de l'objet désiré.

2. Test sur la détection de contours par level set avec initialisation à l'intérieure de l'objet désiré.

3. Test sur des images qui contiennent un seul objet ou plusieurs.

#### Image.1: Image réelle



600 itérations

160 itérations



680 itérations



780 itérations





#### 800 itérations







*Figure 3.4.* Résultats obtenus de l'image.1 avec initialisation à l'extérieur de l'objet d'intérêt.

La figure (3.4) donne les résultats de la segmentation par level set sans réinitialisation de la fonction de distance, qui nous permet de détecter deux objets en même temps avec :

- Le nombre d'itérations utilisées = 800 itérations.

- Le temps d'exécution de l'algorithme ne dépasse pas une minute.

#### Image.2: Image synthétique



*Figure 3.5.* Résultats obtenus de l'image.2 avec initialisation à l'intérieure de l'objet d'intérêt.

Lorsque l'image est simple, nous remarquons que :

- L'expansion du contour est très rapide par apport à limage précédente.
- Le nombre d'itération utilisée est nettement inferieur (130 itérations).

Image .3: Image synthétique à plusieurs objets



Figure 3.6. Résultats obtenus de l'image.3.

Lorsque l'image contient plusieurs objets, nous remarquons que :

- Le rétrécissement du contour de l'image.3 est rapide par apport à l'image.1, car le contour initial choisi est plus proche aux objets d'intérêts.

- Le rétrécissement du contour de l'image.3 est lent par apport à l'image.2, vu la présence de plusieurs objets d'intérêts.

- Le nombre d'itération utilisée est nettement inferieur (300 itérations).

Image.4: Image IRM



Figure 3.7. Résultats obtenus de l'image.4.

Lorsque l'image présente une topologie plus complexe, nous remarquons que :

- L'expansion du contour est lente par apport à l'image synthétique (image.2).

- Le nombre d'itération utilisée est plus grand par apport à l'image.2 (300 itérations).

# 3.6 Conclusion:

A travers ce chapitre nous avons présenté les résultats obtenus par l'application d'algorithme d'ensemble des niveaux sans réinitialisation sur les différents types d'images (Synthétique, IRM et réelle).

Les résultats obtenue au niveau de la simulation sous Matlab montrent l'efficacité de l'algorithme, ceci nous permet donc de passé à la description des différentes parties de notre architecture pour une implémentation Hardware sur un circuit FPGA de Xilinx.

# 4.1 Introduction

Considéré comme l'un des meilleurs outils offerts sur le marché, le circuit FPGA (Field **P**rogrammable **G**ate **A**rray) de Xilinx (voir annexe) offre une panoplie de composants, ainsi qu'un système de développement ISE (*Integrated Software Engineering*) *Fondation* offrant d'énormes possibilités de développement (synthèse, simulation et implémentation automatique).

Dans ce chapitre, il sera question de simulation et implémentation de l'architecture proposée.

La méthodologie de base pour la conception sur circuit FPGA de Xilinx se décompose en quatre étapes:

- 1. Description de l'architecture.
- 2. Simulation fonctionnelle.
- 3. Synthèse et implémentation.
- 4. Simulation temporelle et vérification.

L'architecture que nous avons réalisée a été conçue dans l'environnement ISE 7.2 de Xilinx. Le processus d'implémentation convoite certaines phases de vérification. En premier lieu, l'architecture doit être décrite en langage VHDL (Very High Description Language). Après la description VHDL de l'architecture et sa compilation, une simulation fonctionnelle est introduite par le simulateur (ISE *simulator*), dont les résultats figurent sous forme de chronogrammes. Une fois que la simulation fonctionnelle est validée, une phase de synthèse va permettre de donner un rapport bien détaillé sur les ressources du circuit FPGA consommées et sur la fréquence maximale de fonctionnement. Finalement, une phase de placement et routage éclate schématiquement la surface consommée par l'architecture globale et configure les routages d'interconnexion entre les différents blocs des ressources du circuit FPGA. Dans la simulation temporelle, on vérifie si le circuit obtenu respecte les contraintes temporelles et utilise les délais des portes et les délais dans les interconnexions pour calculer la vitesse maximale. La figure 4.1 regroupe toutes les étapes d'une conception sur circuit FPGA.



Figure 4.1. Schéma des étapes de conception.

# 4.2 Description de l'architecture globale

L'architecture que nous avons proposée est composée de quatre blocs définis comme suit:

- 1. Bloc mémoire.
- 2. Bloc laplacien.

- 3. Bloc courbure.
- 4. Bloc énergie.

L'architecture globale est décrite en figure 4.2 :



Figure 4.2. Architecture globale.

### 4.2.1 Bloc mémoire

Ce bloc est constitué de:

- 1) ROM 1.
- 2) ROM 2.
- 3) Compteur.

Le bloc mémoire consiste, dans un premier temps à charger dans la ROM 1 de taille (1024\*13) toutes les positions des pixels du contour initial avec ses points adjacents à partir d'un fichier texte, la sortie de la ROM1 adresse la ROM2 dont la taille est de 8\*4544, qui contient l'intensité de tous les pixels de notre image, l'adressage de la ROM1 est assuré par la sortie du compteur (scont).

L'architecture de ce bloc est illustrée dans la figure 4.3 :



Figure 4.3. Architecture du Bloc mémoire.

## 4.2.2 Bloc Laplacien

L'opération de laplacien est réalisée par :

- 1. Bloc mémoire.
- 2. Multiplexeur (MUX9-1)
- 3. Multiplieur (mult).
- 4. Accumulateur (acc).
- 5. Compteur.

L'opération de laplacien est réalisée par l'intermédiaire des blocs illustrés sur la figure 4.4, c'est une opération qui nécessite une multiplication entre la sortie de notre bloc mémoire (spo) et la sortie du multiplexeur (outpout) dont la sélection des coefficients Ci ( i allant de 1 jusqu'à 9) [28] est assurée par un compteur. Les premiers neufs résultats du multiplieur (s) sont accumulés par un accumulateur (q), ce résultat généré représente la valeur finale du laplacien du point contour. Le même processus de calcul sera répété jusqu'au dernier point de notre contour initial.

### Remarque

La commande (aclr) remet à zéro l'accumulateur après chaque neuf multiplications.



Figure 4.4. Architecture du Bloc laplacien.

### 4.2.3 Bloc courbure

L'implémentation de la courbure k=  $\nabla(\frac{\nabla \varphi}{|\nabla \varphi|}) = \frac{\varphi_{xx} \varphi_y^2 - 2\varphi_{xy} \varphi_x \varphi_y + \varphi_{yy} \varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}}$  est réalisée par plusieurs blocs :

- 1. Bloc réglage\_ décalage.
- 2. Bloc de calcul des différentes dérivées.
- 3. Bloc de calcul des termes du numérateur de la courbure k.
- 4. Bloc de calcul des termes du dénominateur de la courbure k.



Figure 4.5. Architecteur du bloc courbure.

### a. Bloc réglage\_ décalage

Ce bloc est donné par la figure 4.6, permet du régler le problème de synchronisation des données générées par le bloc mémoire (intensité des points contours initial et leurs points adjacents) afin d'éviter les erreurs pour le calcul des différentes dérivées telle que:  $\phi_x$ ,  $\phi_{xx}$ ,  $\phi_{xy}$ ,  $\phi_y$ ,  $\phi_{yy}$ .

La sortie du bloc mémoire est introduite à l'entrée d'un démultiplexeur (spo) dont la ligne de sélection est assurée par un compteur, les neuf sorties du démultiplexeur sont affectés au bloc registre qui permet la synchronisation de ses sorties (ss, ss1, ss2, ss3, ss4, ss5, ss6, ss7, out8).



#### Figure 4.6. Bloc réglage\_ décalage.

#### b. Bloc de calcul des différentes dérivées

#### *b.1. Calcul de* $\phi_x$

Le bloc de la figure 4.7, permet de calculer la première dérivée de  $\phi$  par rapport à x,  $\Phi_{\chi} = \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2}$ . Cette dérivée est réalisée par le biais des sorties du bloc réglage\_ décalage, tout en faisant une soustraction entre le pixel6 et le pixel4, ensuite un décalage à droite pour réaliser la division par deux.



*Figure 4.7.* Architecture pour le calcul de  $\phi_x$ .

### *b.2. Calcul de* $\varphi_y$

Ce bloc représenté sur la figure 4.8 permet de calculer la première dérivée de  $\phi$  par rapport à y,  $\Phi_y = \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2}$ . Cette dérivée est réalisée par le biais des sorties du bloc réglage\_décalage, tout en faisant une soustraction entre le pixel8 et le pixel2, ensuite un décalage à droite pour réaliser la division par deux.


*Figure 4.8.* Architecture pour le calcul de  $\phi_y$ .

### *b.3. Calcul de* $\phi_{xx}$

Ce bloc représenté sur la figure 4.9 permet de calculer la deuxième dérivée de  $\phi_x$  par rapport à x ,  $\Phi_{xx} = \frac{\Phi_{i+1,j}-2\Phi_{i,j}+\Phi_{i-1,j}}{4}$ . Cette dérivée est réalisée par le biais des sorties du bloc réglage\_décalage, tout en faisant une addition entre le pixel6 et le pixel4, le résultat de cette addition est introduit ensuite à l'entrée d'un soustracteur avec la valeur (2\*pixel5) réalisé par un registre à décalage à gauche, ce résultat sera décalé à droite par deux bits.



*Figure 4.9.* Architecture pour le calcul de  $\phi_{xx}$ .

### *b.4. Calcul de* $\phi_{yy}$

Ce bloc représenté sur la figure 4.10 permet de calculer la deuxième dérivée de  $\phi_y$  par rapport à y ,  $\Phi_{yy} = \frac{\Phi_{i,j+1}-2\Phi_{i,j}+\Phi_{i,j-1}}{4}$ .Cette dérivée est réalisée par le biais des sorties du bloc réglage\_décalage, tout en faisant une addition entre le pixel8 et le pixel2, le résultat de cette addition est introduit ensuite à l'entrée d'un soustracteur avec la valeur (2\*pixel5) réalisé par un registre à décalage à gauche, ce résultat sera décalé à droite par deux bits.



*Figure 4.10:* Architecture *pour le calcul de*  $\phi_{yy}$ .

### b.5. Calcul de $\phi_{xy}$

Ce bloc représenté sur la figure 4.11 permet de calculer la deuxième dérivée de  $\phi_x$  par rapport à y,  $\Phi_{xy} = \frac{\Phi_{i+1,j+1} - \Phi_{i-1,j+1} - \Phi_{i+1,j-1} + \Phi_{i-1,j-1}}{16}$ . Cette dérivée est réalisée par le biais des sorties du bloc réglage\_décalage, tout en faisant une addition entre le pixel7 et le pixel3 et une addition entre le pixel9 et le pixel1, le résultat de ces deux additions sont introduits à l'entrée d'un soustracteur, le résultat de cette soustraction sera décalé à droite par quatre bits.



*Figure 4.11 :* Architecture *pour le calcul de*  $\phi_{xy}$ .

# c. Bloc de calcul des termes du numérateur de la courbure k c.1 Calcul du $\phi_x^2$ et $\phi_y^2$

Le calcul de ces termes nécessite l'utilisation d'un multiplieur et d'un registre pour conserver les résultats finaux comme le représente les figures 4.12 et 4.13.



### *Figure 4.12.* Architecture pour le calcul de $\phi_x^2$ .



*Figure 4.13.* Architecture pour le calcul de  $\phi_y^2$ .

# c.2 Calcul de $\phi_{yy}\phi_x^2$ et de $\phi_{xx}\phi_y^2$

Pour obtenir les valeurs de ces termes, on utilise les résultats précédents suivis d'un multiplieur comme le représente les figures 4.14 et 4.15.



Figure 4.14. Architecture pour le calcul de  $\varphi_{yy}\varphi_x^2$  .



Figure 4.15. Architecture pour le calcul de  $\phi_y^2 \phi_{xx}$ .

## c.3 Calcul du terme $2\phi_{xy}\phi_x\phi_y$

Afin d'obtenir le résultat du dernier terme du numérateur, nous avons utilisé les calculs des termes précédents  $\varphi_{xy}$ ,  $\varphi_x et \varphi_y$ , avec deux multiplieurs de taille différents et un registre à décalage à gauche, comme le schématise la figures 4.16.



*Figure 4.16*. Architecture pour le calcul de  $2\phi_{xy}\phi_x\phi_y$ .

c.4 Calcul de l'addition des termes  $\phi_{yy}\phi_x^2$  et  $\phi_{xx}\phi_y^2$  et concaténation

La somme de ces deux termes est calculée à l'aide d'un additionneur, le résultat est concaténé à droite avec deux bits. La figure 4.17 représente l'architecture faisant ces calcules.



Figure 4.17. Addition et concaténation avec deux bits.

#### c.5 Calcul du numérateur de la courbure k

A partir des résultats calculés précédemment, le calcul du numérateur nécessite la soustraction du terme  $\phi_{yy}\phi_x^2 + \phi \phi_{xx}\phi_y^2$  avec  $2\phi_{xy}\phi_x \phi_y$ , comme le montre la figure 4.18.



Figure 4.18. Bloc du calcul du numérateur de la courbure.

### d. Calcul du dénominateur de la courbure k

Ce bloc nécessite l'implémentation de l'équation  $(\phi_x^2 + \phi_y^2)^{3/2}$ . Pour calculer la valeur de cette équation, nous avons calculé en premier lieu la somme de  $\phi_x^2$  et de  $\phi_y^2$ , ensuite cette somme et multipliée par sa racine carré.



Figure 4.19. Bloc du calcul du dénominateur de la courbure.

## 4.3 Résultats de la simulation

Les étapes suivies dans ce travail pour l'implémentation de l'algorithme de Level-Set sans réinitialisation se résument comme suit:

- Génération des coordonnées des points du contour initial et leurs points adjacents ainsi que les intensités de l'image à tester.
- Chargement des coordonnées des points contours et leurs points adjacents dans la ROM1.
- > Chargement des intensités de l'image à tester dans la ROM2.
- Description de l'architecture en langage VHDL sous l'outil ISE 7.2 et la simulation par l'outil ISE simulator.
- Synthèse de l'architecture.
- Implémentation de l'architecture sur le circuit XC2V8000\_4ff 1152 de la famille VIRTEX-II.
- $\triangleright$

### 4.3.1 Résultat de la simulation du Bloc mémoire

Le résultat de la simulation fonctionnelle du Bloc mémoire est représenté sur la figure

4.20



Figure 4.20: Chronogrammes du bloc mémoire.

# 4.3.2 Résultats de la simulation du Bloc laplacien

Le résultat de la simulation fonctionnelle du Bloc laplacien est représenté sur la figure

4.21.

Now: 10200 ns		0 ns	1 1	505		1010 ns	1 1	1:	515   I	Ĩ	20201	ns I	-
🔊 cik	1												
🔊 acir	0												
SII ce	1												
👌 cikena	1												
표 💦 spo[7:0]	128	( 0 X 192 X 185			160	X 139	192						
🗉 💦 output[3:0]	1	0	X 1	X O	$\chi$ 1	12	$\chi$ 1		(1	X	0		
표 💦 s[11:0]	0		0		192	0	185	3456	(139	X O	192	X	0
표 💦 scont[9:0]	47	0 1	χ 2	Х 3	4	5	X 6	$\left( \begin{array}{c} 7 \end{array} \right)$	8	X 9	_X	10	
🕀 💦 q[11:0]	4032	K	0	I	)	( 1	92	377	3833	X	3972	_X	68

Figure 4.21. Chronogrammes du bloc laplacien.

*Remarque* Les valeurs figurées sur le chronogramme ci-dessus des coefficients (output), la sortie du multiplieur (s) et le résultat de l'accumulateur (q) sont représentés en complément à deux convertis en décimal.

# 4.3.3 Simulation du numérateur de la courbure

Le résultat de la simulation fonctionnelle du Bloc numérateur de la courbure est représenté sur la figure 4.22 :



Figure 4.22 Chronogramme du numérateur de la courbure.

# 4.4 Synthèse et implémentation

Cette étape est une synthèse des différents blocs réalisée de l'architecture globale. Ils sont exécutés par l'outil de synthèse *XST* et permet d'avoir une estimation du taux d'occupation et du temps d'exécution. Ensuite, nous passons à l'implémentation qui est la dernière étape dans la conception. Elle se fera par l'outil *IMPLEMENT DESIGN*. Ce dernier permet aussi de créer un fichier résultat qui donne des informations sur les ressources utilisées (nombre de SLICES, de CLBS, etc.), le routage et le délai d'exécution.

### 4.4.1 Résultats de l'implémentation

Ces résultats nous informent sur les ressources utilisés du circuit XC2V8000\_4ff 1152.

#### a. Bloc memoire

Number of BUFGMUXs	1 out of 16	6%
Number of External IOBs	46 out of 824	5%
Number of LOCed IOBs	0 out of 46	0%
Number of SLICEs	1540 out of 46592	3%

Tableau 4.1. Résultats de l'implémentation du bloc mémoire.



Figure 4.23. Implémentation du bloc mémoire.

## b. Bloc laplacien

Number of BUFGMUXs	1 out of 16	6%
Number of External IOBs	46 out of 824	7%
Number of LOCed IOBs	0 out of 46	0%
Number of SLICEs	1572 out of 46592	3%

Tableau 4.2. Résultats de l'implémentation du bloc laplacien.



Figure 4.24. Implémentation du bloc laplacien.

#### c. bloc courbure

### c.1 bloc numérateur

Number of BUFGMUXs	1 out of 16	6%
Number of External IOBs	353 out of 824	42%
Number of LOCed IOBs	0 out of 353	0%
Number of SLICEs	16404 out of 46592	35%

Tableau 4.3 Résultats de l'implémentation du bloc numérateur.



Figure 4.25. Implémentation du bloc numérateur.

# 4.5 Conclusion

Dans ce chapitre, nous avons présenté la méthodologie de conception de notre architecture. Nous avons également donné, les résultats de simulation ainsi que les résultats d'implémentation obtenue sur le circuit XC2V8000\_4ff 1152 de la famille VIRTEX-II. Pour cela, nous sommes passés par plusieurs étapes, chacune d'entre elles utilise un outil bien spécifique.

# Liste des figures

# Chapitre1 : Généralité sur traitement d'image et segmentation

Figure 1.1. Le pixel	3
Figure 1.2. Système de traitement numérique d'images	6
Figure 1.3. Histogramme de répartition pour trois classes	9
Figure 1.4. Segmentation par division	10
Figure 1.5. Segmentation par division	10
Figure 1.6. Principe de la division-fusion	11
Figure 1.7. Quelques profils de contour (a) : marche, (b) : rampe, (c) :toit	12
Figure 1.8. Allure des dérivées premières et seconde d'une transition de type	
' saut d'amplitude'	13
Figure 1.9. Orientation du contour	14
Figure 1.10. Applications des détecteurs de Prewitt et Sobel	15
Figure 1.11. Opérateur MDIF	17
Figure 1.12. Domaines de nagao	18
Figure 1.13. Opérateur NAGDIF	18
Figure 1.14. Opérateurs du deuxième ordre	20
Figure 1.15. Exemple de dilatation sur une image binaire	22
Figure 1.16. Exemple de dilatation sur une image à plusieurs niveaux de gris	23
Figure 1.17. Exemple d'érosion sur une image binaire	24
Figure 1.18. Exemple de l'érosion sur une image à plusieurs niveaux de gris	25
Figure 1.19. Exemple d'ouverture sur une image binaire	26
Figure 1.20. Exemple d'ouverture sur une image à plusieurs niveaux de gris	26
Figure 1.21. Exemple de fermeture sur une image binaire	26
Figure 1.22. Exemple de fermeture sur une image à plusieurs niveaux de gris	26
Figure 1.23. Opérateurs morphologiques	28
Figure 1.24. Principe des contours actifs	30

# Chapitre 2 : Les modèles déformables

Figure 2.1. Principe d'un contour actif appliqué à une image coronarographique	31
Figure 2.2. Evolution de la courbe C	7
Figure 2.3. Changement de topologie au cours De l'évolution de la courbe	38

# Chapitre 3 : Simulation de l'application sous Matlab.

Figure 3.1. Chaine de traitement49
Figure 3.2.Quelques Images de la base source50
Figure 3.3. Initialisations du contour51
Figure3.4. Résultats obtenus de l'image.1 avec initialisation à l'extérieur de l'objet
d'intérêt53
Figure3.5.Résultats obtenus de l'image.2 avec initialisation à l'intérieure de
l'objet .d'intérêt
Figure 3.6. Résultats obtenus de l'image. 354
Figure 3.7. Résultats obtenus de l'image.455

# **Chapitre4 : Conception et implémentation**

Figure 4.1. Schéma des étapes de conception58
Figure 4.2. Architecture globale59
Figure 4.3. Architecture du Bloc mémoire60
Figure 4.4. Architecture du Bloc laplacien61
Figure 4.5. Architecteur du bloc courbure62
Figure 4.6. Bloc réglage_ décalage62
Figure 4.7. Architecture pour le calcul de $\phi_x$ 63
Figure 4.8. Architecture pour le calcul de $\phi_y$ 64
Figure 4.9. Architecture pour le calcul de $\phi_{xx}$ 65
Figure 4.10. Architecture pour le calcul de $\phi_{yy}$ 66
Figure 4.11 . Architecture pour le calcul de $\varphi_{xy}$ 67
Figure 4.12. Architecture pour le calcul de $\varphi_x^2$ 67
Figure 4.13. Architecture pour le calcul de $\phi_y^2$
Figure 4.14. Architecture pour le calcul de $\phi_{yy}\phi_x^2$ 68
Figure 4.15. Architecture pour le calcul de $\phi_{xx} \phi_y^2$ 69
Figure 4.16. Architecture pour le calcul de $2\varphi_{xy}\varphi_x \varphi_y$ 69
Figure 4.17. Addition et concaténation avec deux bits
Figure 4.18. Bloc du calcul du numérateur de la courbure70

Figure 4.19. Bloc du calcul du dénominateur de la courbure	71
Figure 4.20. Chronogramme de bloc mémoire	.72
Figure 4.21. Chronogramme du Bloc laplacien	72
Figure 4.22.Chronogramme du numérateur de la courbure	.73
Figure 4.23. Implémentation du bloc mémoire	74
Figure 4.24. Implémentation du bloc laplacien	.74
Figure 4.25. Implémentation du bloc numérateur	75

# Liste des tableaux

# Chapitre 4:Conception et implémentation

Fableau 4.1. Résultats de l'implémentation du bloc mémoire	73
rableau 4.2. Résultats de l'implémentation du bloc laplacien	74
rableau 4.3 Résultats de l'implémentation du bloc numérateur	75

# Table des matières

Introduction Générale

# Chapitre 1 Généralité sur traitement d'image et la segmentation.

1.1 Introduction1
1.2 Définition de l'image1
1.3 Image numérique2
1.4 Les caractéristiques d'une image numérique2
1.4.1 Pixel2
1.4.2 Dimension
1.4.3 Bruit
1.4.4 Histogramme3
1.4.5 Contours et textures4
1.4.6 Luminance4
1.4.7 Contraste4
1.4.8 Image a niveaux de gris5
1.4.9 Image en couleurs5
1.5 Les systèmes de traitement d'images6
1.5.1 Acquisition des données images6
a Dispositifs de numérisation d'images6
1.5.2 Le prétraitement7
a Filtrage numérique7
1.6 La segmentation7
1.6.1 Approche région
a Segmentation par classification8
b Segmentation par division (split)9
c Approche fusion10
d Approche par division-fusion10
1.6.2 Approche frontière11
a Introduction11
b Les méthode dérivatives12
b.1 Opérateurs dérivatifs du premier ordre14

b.1.1 Operateurs de Prewitt et de Sobel	15
b.1.2 Opérateurs de gradient directionnel de Kirsh	16
b.1.3 Opérateurs MDIF	16
b.1.4 OpérateurNAGDIF	17
b.2 Operateurs dérivatifs du deuxième ordre	19
b.2.1 Autres operateurs du deuxième ordre	20
c La morphologie mathématique	21
c.1 Transformations morphologiques	21
c.1.1 Principes généraux	21
c.1.2 Elément structurant	22
c.2 Opérateurs fondamentaux	22
c.2.1 La dilatation	22
c.2.1.1 Dilatation des images binaire	22
c.2.1.2 Dilatation des images à plusieurs niveaux de gris	23
c.2.2 L'érosion	23
c.2.2.1 Erosion des images binaires	24
c.2.2.2 Erosion des images à plusieurs niveaux de gris	24
c.3 Ouverture et fermeture	25
c.4 Autres opérateurs	27
c.4.1 Gradient morphologique (ou Gradient symétrisé)	27
c.4.2 Gradient par dilatation	27
c.4.3 Gradient par érosion	28
c.4.4 Laplacien morphologique	28
d Interprétation des résultats	28
e Les contours actifs	29
e.1 Définition	29
1.7 Conclusion	30

Introduction	31
2.2 Les contours actifs paramétriques	.32
2.2.1 Les contour actifs traditionnels	32

2.2.2 Conclusion35	
2.3 Les contours actifs Géométriques35	
2.3.1 Introduction35	
2.3.2 Méthode d'ensemble de niveau zéro (Level sets traditionnels)	
a Construction de la fonction de distance41	
b Réinitialisation de la fonction de distance42	
2.3.3 Formulation variationnelle des levels sets sans la réinitialisation de la fonction de	
distance43	
a Initialisation flexible de la fonction d'ensemble de niveaux46	
2.4 Conclusion47	
Chapitre 3 Simulation de l'application sous Matlab.	
3.1 Introduction	
3.2 L'outil de développement48	
3.3 Description de l'algorithme de contour actif géométrique sans réinitialisation	
de la fonction de distance49	
3.4 Méthodologie de travail49	
3.4.1 Ensemble d'image testé50	
3.4.2 Prétraitement	
3.4.3 Initialisation	
a. Résultats d'initialisation51	
3.4.4 Calcule des énergies51	
3.4.5 Evolution de la courbe initiale51	
3.5 Application d'algorithme sur les déférents types d'image52	
3.6 Conclusion	
Chapitre4 Conception et Implémentation.	

4.1	Introduction	57
4.2	Description de l'architecture globale	.58
4.2.1	Bloc mémoire	59
4.2.2	Bloc Laplacien	.60
4.2.3	Bloc courbure	61
a. Blo	ccréglage_decalage	62
b. B	oc calcul des différentes drivées	.63

b.1. Calcul de $\varphi_x$	63
b.2. Calcul de $\varphi_y$	63
b.3. Calcul de $\varphi_{xx}$	64
b.4. Calcul de $\phi_{yy}$	65
b.5. Calcul de $\varphi_{xy}$	66
c. Bloc de calcul des termes du numérateur de la courbure k	67
c.1 Calcul du $\varphi_x^2$ et $\varphi_y^2$	67
c.2 Calcul de $\varphi_{yy}\varphi_x^2$ et de $\varphi_{xx}\varphi_y^2$	
c.3 Calcul du terme $2\varphi_{xy}\varphi_x\varphi_y$	69
c.4 Calcul de l'addition des termes $\phi_{yy}\phi_x^2$ et $\phi_{xx}\phi_y^2$ et concaténation	70
c.5 Calcul du numérateur de la courbure k	70
d. Calcul du dénominateur de la courbure k	71
4.3 Résultats de la simulation	71
4.3.1 Résultat de la simulation du Blocs mémoires	72
4.3.2 Résultats de la simulation du Bloc laplacien	72
4.3.3 Simulation du numérateur de la courbure	72
4.4 Synthèse et implémentation	73
4.4.1 Résultats de l'implémentation	73
a. Bloc mémoire	73
b. Bloc laplacien	74
c. bloc courbure	75
4.5 Conclusion	75
Conclusion générale	В
Annexe	C
Bibliographie	D

[1] Ali. Nait, A. Christine, Cavaro Ménard, 'compression des images et des signaux médicaux', ISBN 978-2-7462-1493-4, lavoisier, 2007.

[2] Chaouchi Chahinez: 'Etude et implémentation de détecteurs de contours', Ecole Nationale Polytechnique Département d'Electronique Alger, Septembre 2004

[3] Nour El-Houda Golea :'Tatouage numérique des images couleurs RGB', faculté des sciences de l'ingénieur département d'informatique université ELhadjlakhder-Batna, 2010.

[4] Ben khadidja Fatiha, Mahdjoub Hadia:'segmentation d'image par Level Set', faculté des sciences de l'ingénieur département d'électronique l'université Saad Dahlab Blida, 2006.

[5] René Husson, Claude Lung, Jean-François aubry, Jamal Afouz et Didier Wolf: 'Automatique', Dunod, paris, 2007.

[6] Mestar Amine, Khelifi Riad: 'Segmentation des images coronarographiques Application à la détection des cardiopathies', en vue de l'obtention du diplôme d'ingénieur d'état en électronique école nationale polytechnique Département d'électronique Alger, 2005.

[7] Boulares Abdennour : 'Segmentation des images IRM par les level set', Thèse d'ingénieur d'état en Electronique, Ecole Nationale polytechnique d'Alger, 2008.

[8]R.H Abdeljalil, B.Mourad, B.Samir :'Modélisation d'un Scanner Ultrasonique', Thèse d'ingénieur d'état en Electronique, Universite de colonel el hadj lakhder Batna Alger,2001.

[9] J.-p coquerez, S.philipp : 'Analyse d'image: filtrage et segmentation', Edition Masson, 1995.

[10] M.E. Hyche and Ezquerra : 'Spatiotemporal detection of arterial structure using active contours', Proceedings of SPIE, Vol. 1808, pp. 52-62, 1992.

[11] J.M. Salotti :'Gestion des informations dans les premières étapes de la vision par ordinateur', Thèse de doctorat de l'institut national polytechnique de Grenoble, spécialité informatique, mars 1992.

[12] R.C. Gonzalez,' Digital Image Processing', Second edition, Addison Wesley, 1987.

[13] N. Lassouaoui: 'Segmentation des images biomédicales par des approches fractales, Conception d'un logiciel de traitement et de segmentation des images', Thèse de Magister, Ecole Nationale Polytechnique d'Alger, 2000.

[14] L. Cuvillon: 'Compensation du battement cardiaque en chirurgie robotisée: Asservissement visuel d'un robot médical avec flexibilité', Thèse de doctorat, l'Université Louis Pasteur Strasbourg I, décembre 2006.

[15] Ammour Nassim:'Méthode de segmentation d'image rapide basée sur l'évolution probabiliste d'un graphe et de la théorie de l'évidence', Thèse de doctorat, Ecole Nationale Polytechnique d'Alger, 2009.

[16] M. Passat: 'Contribution à la segmentation des réseaux vasculaires cérébraux obtenus en IRM. Intégration de connaissance anatomique pour le guidage d'outils de morphologie mathématique', Thèse de doctorat, Université Louis Pasteur de Strasbourg -École Doctorale MSII Laboratoire de Sciences de l'Image, de l'Informatique et de la Télédétection, septembre 2005.

[17] A. Kieffer: 'Travail d'étude et de recherche caractérisation de textures dans les images satellites', Master 1 Informatique fondamentale, Université Louis Pasteur Strasbourg, Année 2005-2006.

[18] N. Serafini: 'Implantation d'une librairie de morphologie mathématique sur un processeur DSP', Ecole d'ingénieurs de Genève HES, décembre 2001.

[19] Yahya Zoubir Bahia: 'Segmentation de l'arbre coronaire à partir de coronographies', Thèse d'ingénieur d'état en Electronique, Ecole Nationale polytechnique d'Alger, 2007. [20] J. Goutsias, L. Vincent and S. Bloomberg: 'Mathematecal morphology and its applications to image and signal processing', Kluwer academic publishers, 2002.

[21] Khelifi Riad: 'Segmentation des images coronarographiques par les contours actifs', Thèse de Magister, Ecole Nationale Polytechnique d'Alger, 2008.

[22] Boudermine Rabeh:'Détection des cavité cardiaque en utilisant les contours actifs', Thèse d'ingénieur d'état en informatique, Ecole National Supérieure d'Informatique (E.S.I) Oued-Smar, Alger, 2009.

[23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes:Active Contour models," Int'l J. Computer Vision, vol.1, pp. 312-333, 1988.

[24] S.Osher, J.A. Sethian: 'Fronts propagating with curvature-dependant speed: Algorithms based on Hamilton-Jacobi formulations', Journal of Computational Physics, vol. 79(1), pp. 12-49, 1988.

[25] M. Sussman, and E. Fatemi: 'An Efficient, Interface- Preserving Level Set Re-distancing Algorithm and its Application to interfacial Incompressible Fluid Flow', preprint, 1995.

[26] V. Caselles, R. Kimmel, and G. Sapiro: "On geodesic active contours,"Int. J. Comput. Vis., vol. 22, no. 1, pp. 61–79, 1997.

[27] C. Li, C. Xu, C. Gui, and M.D. Fox: 'Level set Evolution without re-initialization: a new variational formulation', IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 430-436, San Diego, 2005.

[28] H. DEBA, B. DAGUIANI : 'Implémentation d'un détecteur de contour sur FPGA, Méthode de CANNY-DERICHE' faculté des sciences de l'ingénieur département d'électronique l'université Saad Dahlab Blida, 2011.

# Présentation des FPGA et conception sur ISE 7.2

## 1. Description d'un FPGA (Field Programmable Gate Array)

Un FPGA ou (réseaux logiques programmables) consiste en une matrice carrée de cellules configurables CLB (Configurables Logique Blocs) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs d'entrées /sorties IOB (Input Output Blocs) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs. La figure 1 présente l'architecture générale d'un FPGA.



Figure 1. Architecture générale d'un FPGA.

### 2. Les familles des FPGA's

Grâce aux évolutions de la technologie microélectronique, les FPGAs deviennent de plus en plus performants avec des capacités d'intégration, sans cesse augmentées. Longtemps réalisées autour de blocs logiques configurables à base de LUT (Look Up Table), les récentes familles FPGAs(VIRTEX-II ,VIRTEX-II pro) peuvent aujourd'hui comporter des mémoires de 18 kbits, de blocs multiplieurs et processeurs.

Les circuits FPGA de XILINIX sont caractérisés par une nomenclature spécifique qui définit les performances de chaque famille. Cette nomenclature est la suivante:



Figure 2: Nomenclature d'un FPGA de XILINX.

2V: "Device type" représente le type de famille, dans ce cas est: VIRTEX-II.

-6: "Speed Grade" représente la vitesse du composant qui dépend de la technologie du circuit.

- **1152**: "Number of pins "représente le nombre de pins.
- X: est la lettre initiale de Xillinx.
- **F**: "Package type" représente le type de boîtier.
- C: "Temperature Range" représente la gamme de température tolérée.

### 3. La famille VIRTEX-II

La famille VIRTEX-II a été conçue pour réaliser des conceptions à faible ou grande densité d'intégration et qui exigent des performances élevées (elle peut atteindre jusqu'à 10 millions de portes logiques). La fréquence de son utilisation peut être portée à 420 Mhz. Actuellement, cette famille est utilisée dans plusieurs applications tels que: les réseaux, la télécommunication, la vidéo et les applications DSP (Digital Signal Processing).

L'architecture d'un circuit VIRTEX II est montrée sur la figure 3. Celle-ci se présente comme un réseau de bloc logique configurable. Ces derniers sont des blocs Input/Output (IOB) et des blocs logiques configurables (CLB).



Figure 3: Architecture interne d'un VIRTEX- II.

## **3.1** Le bloc logique interne configurable

La logique interne configurable inclut quatre éléments majeurs organisés sous une forme régulière et un chemin dédié à la propagation de la retenue. Les éléments en question sont:

- Bloc Logique Configurable (CLB).
- Bloc Select RAM 18 K bits à doubles ports.

- > Bloc multiplieur (18bit x 18bit).
- DCM (Digital Clock Manager).

La connexion entre tout ces blocs est assurée d'une manière programmable grâce à des ressources d'interconnections configurables GRM " General Routing Matrix ".

## 3.2 Bloc d'entrées/sorties (IOB)

Les blocs d'entrée/sortie fournissent une interface entre les broches externes du circuit et la logique interne. Deux IOBs peuvent être utilisés comme une paire différentielle. Celle-ci est souvent connectée à une même matrice d'interconnexion .Comme cela est illustrée sur la figure suivante.



Figure 4: Connexion des IOBs à une matrice d'interconnexion.

Les IOBs sont configurés soit en : entrée, sortie, bidirectionnels ou en haut impédance. Ils incluent six bascules D, leur utilisation est souvent recommandée pour la synchronisation des signaux de communication. Chaque bascule peut être configurée comme étant:

- Sensible au front montant ou descendant d'une horloge.
- Sensible au niveau d'une horloge.



Figure 5: Bloc IOB de VIRTEX II.

Leur utilisation suivant les trois états, l'une ou les deux bascules peuvent fournir un chemin à double débit. Ceci est réalisé par l'utilisation de deux horloges déphasées de 180° et générées par une DCM. Cela est montré dans la figure.



Figure 6: Registre à double débit.

## 3.3 Bloc logique configurable (CLB)

Les CLBs sont implémentés sous forme d'une matrice (N\*M, Ils incluent toute la logique nécessaire pour la conception des circuits combinatoires et séquentiels. Le CLB était l'élément déterminant des performances des premiers circuits FPGAs. Mais ce n'est plus le cas pour le VIRTEX-II, où on ne parle plus de CLB mais de Slice. Chaque CLB est composé de quatre Slices. Leur connexion au routage global est assurée par les matrices d'aiguillages GRM.

Les quatre slices sont implémentés en deux paires sur deux colonnes, chaque paire possède un chemin rapide dédié à la propagation de la retenue, leur interconnexion est assurée par un signal de liaison commun.



Figure 7: CLB de VIRTEX II.

## 3.3.1. Description d'un Slice

L'architecture d'un Slice se présente comme une structure symétrique. Cette caractéristique rend ce dernier flexible à l'optimisation des applications données. Chaque Slice est composé de:

- > Deux générateurs de fonction F et G à quatre entrées et une sortie.
- Deux éléments de stockage (bascule D).
- > Des portes logiques (XOR) dédiées à l'implémentation des Full Adders.
- > Une logique pour la propagation de la retenue.

#### > Des multiplexeurs.



Figure 8: Schéma détaillé de la moitié supérieure d'un Slice.

Les générateurs de fonction peuvent être configurés comme: des Look Up Table (LUT) à quatre entrées, des RAMs à 16 bits ou des registres à décalage de 16 bits.

Les sorties des générateurs de fonction sont considérées comme étant les sorties du Slice utilisé ou Les entrées des deux bascules (D).

### 3.4 Bloc Select RAM à 18-Kbits

Le Virtex II inclut dans son architecture un nombre important de blocs mémoires de 18 Kbits. Ceci augmente l'espace mémoire du circuit, si les CLBs sont configurés comme des mémoires. Chaque bloc SelectRAM est une RAM à doubles ports de taille 2\*18-Kbits. Ces derniers peuvent avoir indépendamment les signaux d'horloges et de commandes. D'une manière générale, les blocs SelectRAM sont implémentés suivant l'une des trois configurations suivantes :

- Lire après écriture (Read After Write);
- Lire avant écriture (Read Before Write);
- Lire seulement (Read Only).

Ces blocs mémoires peuvent être utilisés comme des RAMs à simple ou double port, et ils supportent plusieurs configurations. Ces configurations sont déterminées par un compromis entre la taille de la donnée et le nombre d'adresses lignes. Le compromis en question doit respecter la taille globale de la mémoire qui est de 18-kbits.

### 3.5 Bloc Multiplieur 18 bits\*18 bits

La famille Virtex II incorpore des blocs Multiplieurs 18 bits\*18bits. L'implémentation d'un multiplieur sur cette famille peut être effectuée de deux manières.

La première est l'utilisation directe des blocs multiplieurs. La seconde est la configuration des CLBs en multiplieurs. Cette tâche est réalisée grâce à un outil **Coregen IP** qui nous permet de générer des modules pré-optimisés. La seconde est jugée moins efficace par rapport à la première, vu que l'utilisation des CLBs présente une consommation supplémentaire d'énergie.

La disposition des blocs multiplieurs se trouve implémentée juste à côté des blocs SelectRAM. Cette technique permet d'augmenter les performances d'une application si les opérandes proviennent d'une mémoire.

Ces blocs multiplieurs permettent d'effectuer une multiplication sur des opérandes signés ou non signés et avoir un résultat sur 36 bits, ils peuvent aussi avoir des entrées sorties registrées.

Chaque Bloc SelectRAM et un bloc multiplieur sont connectés au routage global grâce à quatre matrices d'interconnexion.

### 3.6 Les DCMs

Le bloc DCM (Digital Clock Manager) fournit des solutions auto-étalonnées, pour la compensation du temps de l'horloge, la multiplication de l'horloge et sa division, c'est-à-dire, possibilité de diminuer le temps de montée d'une horloge et d'éliminer les pertes dues aux routages. Le DCM utilise un routage particulier dédié qui permet une bonne précision pour le contrôle de l'horloge et de la fréquence. Le routage en question est composé par huit lignes par cadran.

Les DCMs sont situés au-dessus et au-dessous de chaque bloc Select Ram et multiplieur.

### 3.7 Routage global et interconnexion

Les ressources de routage global d'un circuit sont optimisées pour d'éventuelles prévisions des performances d'une application donnée. L'accès au routage des différents blocs disponibles sur le circuit (IOB, CLB, Bloc RAM, Multiplieur, et les DCMs) est réalisé d'une manière programmable grâce à des matrices d'aiguillage. Cela est montré sur la figure-9-.



Figure 9: L'interconnexion des ressources interne du circuit FPGA

## 4. Techniques de programmation des FPGA

Les circuits FPGA ne possèdent pas de programme résident. A chaque mise sous tension, il est nécessaire de les configurer. Leur configuration permet d'établir des interconnexions entre les CLB et les IOB. La configuration du circuit est mémorisée sur la couche réseau SRAM à partir d'une EPROM externe. En effet, un dispositif interne permet à chaque mise sous tension de charger la SRAM interne à partir de l'EPROM.

Ainsi on conçoit aisément qu'un même circuit puisse être exploité successivement avec des EPROMs différentes puisque sa programmation interne n'est jamais définitive.

Le format des données du fichier de configuration est produit automatiquement par le logiciel de développement sous forme d'un ensemble de bits organisés en champs de données.

### 5. Les étapes de conception d'application sur FPGA

La conception, la description et la simulation des composants et systèmes en langage VHDL (Very High Description Langage) présentent deux aspects distincts pour un modèle.

- Une vue externe du circuit: appelée « Spécification ».
- Une vue interne du circuit: appelée « Architecture ».

Chacune de ces deux parties sera une unité de conception et ce sont ces unités qui constituent la bibliothèque VHDL. Une bibliothèque peut compter des centaines d'unités de conception séparées qui permettent de modifier facilement le fonctionnement interne.

Le VHDL est un langage très modulaire, on peut ne pas écrire de longues descriptions, mais des unités plus petites et hiérarchisées; certaines parties de ces descriptions peuvent être compilées séparément; elles sont suffisantes en elles-mêmes pour être comprises. Ce sont les unités de conception. On peut dire que:

- Le circuit est considéré comme une entité (modèle).
- Cette entité peut être composée par d'autres entités.
- Chaque entité est composée de deux parties, interface et corps. Elle permet d'avoir plusieurs corps pour une même interface.

Parmi les avantages du VHDL nous insistons sur les aspects suivants:

- Le langage permet une description hiérarchique: un système peut être modélisé par un ensemble de composants, lequel, à son tour, peut être modélisé par l'intermédiaire d'un ensemble de sous composants.
- Le langage n'est pas spécifique à une technologie particulière mais peut quand même supporter des caractéristiques spécifiques à une technologie cible.

Le VHDL permet des descriptions génériques ou paramétrées. Ces caractéristiques seront largement exploitées dans la description du réseau de neurone.

L'outil de ISE Simulator utilisé est l'un des logiciels de CAO les plus répandus, sur les plateformes de type PC pour la mise en œuvre des circuits programmables FPGA et qui réalisent la simulation.

La conception se passe principalement en trois phases:

- La saisie du circuit.
- L'implémentation.
- La configuration du composant.

Auxquelles il faut ajouter les phases de vérification:

- La simulation fonctionnelle: permet de vérifier la validité du circuit par rapport au cahier des charges, celle-ci ne tient pas compte des capacités de liaison dues au routage entre les différentes cellules.
- La simulation temporelle: vérifie la fonctionnalité du circuit en prenant en compte les longueurs d'interconnexion et les retards apportés par les capacités liées au routage.

### 6. Les outils de conception



Figure 10. Phases de conception sur FPGA

## 6.1. Design entry

Cet outil permet la description d'une architecture ou d'un design soit en code VHDL, soit par l'utilisation d'un éditeur de schéma ou bien par les diagrammes d'état.

## 6.2. Design Synthesis

Cet outil transforme la description VHDL en portes logiques (description proche des ressources matérielles) et procède à l'optimisation: les signaux inutilisés sont retirés, les expressions booléennes sont simplifiées, les signaux équivalents sont détectés.

## 6.3. Design Implementation

Il réalise le mapping, c'est la projection des équations sur les différents blocs de circuit. L'outil place and route assure l'implémentation des différents blocs et définit les chemins d'interconnexions entre ces derniers afin de faciliter le routage.
## 6.4. Static Timing Analysis

Cet outil permet d'effectuer une analyse temporelle d'une architecture. Afin de déterminer ces performances temporelles.

## 6.5. Design Verification

Cet outil nous permet de réaliser une simulation temporelle (vérification réelle) en tenant compte des résultats obtenus par l'outil précédent.