

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
– MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE –
ECOLE NATIONALE POLYTECHNIQUE.



DEPARTEMENT D'ELECTRONIQUE

Projet de fin d'études

**En vue de l'obtention du
Diplôme d'ingénieur d'état en Electronique**

Thème :

**Segmentation par étiquetage des artères
coronaires avec implémentation sur DSP
C6000**

Proposé et dirigé par :

Dr. L.HAMAMI (MC)

Réalisé par:

M. FRADJ Mohammed Karim.

M. LOUKAL Abdelkader

Promotion : Juin 2007

REMERCIEMENTS

Nous adressons nos plus vifs remerciements à notre promotrice Dr L. HAMAMI pour les conseils avisés qu'elle nous a toujours prodigués, pour les connaissances dont elle nous a fait bénéficier, pour son suivi, sa confiance, et surtout l'honneur qu'elle nous a prodigué en encadrant ce travail.

Nous remercions M. D.BERKANI, professeur à l'école nationale polytechnique, de nous avoir fait l'honneur de présider le jury et pour ses encouragements de valeur.

Nos remerciements vont aussi à Mme. A.MOUSSAOUI pour avoir accepté d'examiner ce modeste travail, ainsi qu'au Docteur SAADAOUI, qui a bien voulu nous honorer par sa présence

Nous tenons à remercier également l'ensemble des enseignants qui ont contribué à notre formation.

Nos sincères remerciements vont aussi à toutes les personnes qui, de près ou de loin ont contribué à la réalisation de ce travail, en particulier Riad KHELIFI du Gens-de-Glace, et NADIA du laboratoire signal et communication.

Nous voudrions enfin remercier en particulier la troisième promotion de l'ENPEI ainsi que toute la promotion d'électronique 2007 de l'ENP.

Kader & Karim

DEDICACE

*A mes très chers parents
A mes frères et sœur : Kader, Malhod et Imane
A Mami
A ma promotrice : L. HAMAMI
A binomi : Kader
A mes amis : C'H, Mehmet, Riad, Fateh, Okba, Alilou...
A tous ceux qui ont su croire en moi :
Ma taymour, le Pingouin, Aschenka...
A tous ceux qui me sont chers
Je dédie ce modeste travail*

FMK

DEDICACE

*A mes parents
A mes frères et soeurs
A ma grande famille
A ma promotrice Dr L.Hamami
A mon binôme FMK
Ainsi qu'à tout mes amis...*

LOUKAL

ملخص

يقوم هذا العمل بتجزئة صوراً نجيوجرافيا القلب و الهدف من ذلك هو الكشف على تضيق الأوعية الدموية المسببة بالدرجة الأولى لتصلب الشرايين. عملنا ينقسم إلى اربعة أقسام, في الأولى نبحث عن المحيط الأمثل لشبكة الأوعية, هذا البحث سمح لنا بدراسة عدد كبير من كواشف المحيطات . في القسم الثاني نبحث عن هيكل الشبكة بعد القيام بدراسة بعض الخورزميات. في القسم الثالث ندرس عملية وسم الشريان التاجي. نهاية هذا العمل تتضمن تطبيق خورزمية الوسم على DSP C 6211 , من عائلة DSP C 6000, و كذا بعض خوارزميات معالجة الصور الرقمية.

الكلمات المفاتيح: التجزئة , صور الأوعية , المحيط , الهيكل , شبكة الشرايين التاجي, DSP C 6000, الوسم.

Résumé

Ce travail traite de la segmentation d'images d'angiographie cardiaque. Le but de celui-ci est de détecter les lésions vasculaires (sténoses) dues en majeure partie à l'athérosclérose. Il se divise en quatre parties, dans la première, nous recherchons le contour optimal du réseau coronarien, cette recherche a permis de passer en revue tous les types de détecteurs de contours. La seconde partie consiste en l'extraction du squelette du réseau coronarien, nous recherchons le squelette du réseau après avoir fait une étude comparative entre les différents algorithmes de squelettisation. La troisième partie, fait l'objet de l'étiquetage des artères coronaires, une étude théorique est suivie d'algorithmes expliquant le processus d'étiquetage. Le travail est conclu par l'implémentation de l'algorithme d'étiquetage sur le DSP C6211, de la famille C6000, ainsi que de routines de traitement d'images.

Mots clés : segmentation – angiographie cardiaque- étiquetage – contour- squelettisation –réseau coronaire – DSP C6000.

Abstract

This work is about segmentation of images of angiograms, the goal of this last is vascular lesion's detection caused in major part by atherosclerosis. Our work is divided in four parts. In the first part, we search for the vascular net an optimal edge, this lead to make a state of art of edge detectors. In the second part, consists of the net coronary skeleton extraction, we search for the skeleton of the coronary net while doing a comparative study between the different algorithms of skeleton. The third part is about labelling the coronary arteries; a theoretical study is followed by algorithms explaining the labelling process. This work is concluded by implementation of the labelling algorithm on DSP C6211, of the C6000 family, and some routines of image processing.

Key words : segmentation– angiograms –labelling – edge – skeleton – coronary net – DSP C6000.

Table des matières

CHAPITRE I.....	5
I.1 INTRODUCTION	5
I.2 ANATOMIE DU CŒUR, ET DES ARTERES CORONAIRES.....	10
I.3 ACQUISITION DES IMAGES	10
I.3.1 L'ANGIOGRAPHIE PAR RAYONS X.....	10
I.3.1.1 Description	12
I.3.1.2 Formation de l'image.....	12
I.3.1.3 l'émission de rayons X	12
I.3.1.4 Transmission de rayons X	14
I.3.1.5 Détection de rayons X	15
I.3.1.6 Trajectoires d'acquisition et séquences d'images.....	16
I.3.1.7 Mode statique	16
I.3.1.8 Mode rotationnel.....	18
I.4 CONCLUSION	21
CHAPITRE II.....	22
II.1 INTRODUCTION	22
II.2 GENERALITE	22
II.2.1 IMAGE NUMERIQUE	23
II.2.2 L'ACQUISITION DE L'IMAGE	23
II.2.3 PARTITIONS ELEMENTAIRES	24
II.2.5 LA RESOLUTION	24
II.2.6 IMAGES A NIVEAUX DE GRIS	25
II.2.7 HISTOGRAMME	25
II.2.8 IMAGES EN COULEURS.....	26
II.2.9 INTENSITE	26
II.2.10 CONTRASTE	26
II.2.11 LE BRUIT	26
II.2.12 LES SOURCES DE DEGRADATIONS DANS UNE IMAGE.....	27
II.2.12.1 BRUIT LIE AU CONTEXTE DE L'ACQUISITION :	27
II.2.12.2 BRUIT LIE AU CAPTEUR	27
II.2.12.3 BRUIT LIE A L'ECHANTILLONNAGE	28
II.2.12.4 BRUIT LIE A LA NATURE DE LA SCENE	29
II.2.13 VOISINAGE ET METRIQUES ASSOCIEES	30
II.2.14 INTERIEUR ET EXTERIEUR :	30
II.2.15 NEGATIF.....	32
II.2.17 AMELIORATION DE CONTRASTE	33
II.3 FILTRAGE DES IMAGES NUMERIQUES	34
II.3.1 FILTRAGE LINEAIRE.....	35
II.3.1.1 FILTRE MOYENNEUR	36
II.3.1.2. FILTRE SMOOTH	36
II.3.2 FILTRAGE LINEAIRE ADAPTATIF	37
II.3.2.1 GENERALITES.....	37
II.3.2.2 MOYENNE ADAPTATIVE	37
II.3.3 FILTRAGE NON LINEAIRE.....	38
II.3.3.1 FILTRAGE DE NAGAO	41
II.3.4 FILTRAGE MORPHOLOGIQUE	42
II.3.5 FILTRAGE HOMOMORPHIQUE	44
II.4 CONCLUSION.....	44
CHAPITRE III	44
III.1 INTRDUCTION.....	44

III.2 APPROCHE FRONTIERE OU DETECTION DE CONTOURS.....	45
III.2.1 OPERATEUR DERIVATIF DU PREMIER ORDRE.....	46
III.2.1.1 OPERATEURS DE ROBERTS.....	46
III.2.1.2 OPERATEUR DE PREWITT ET DE SOBEL.....	48
III.2.1.3. OPERATEUR MDIF ET NAGDIF.....	49
III.2.1.4 OPERATEURS GRADIENTS DIRECTIONNELS : KIRSH, 3 ET 5 NIVEAUX.....	50
III.2.2 OPERATEURS DERIVATIFS DU SECOND ORDRE.....	50
III.2.3 GRADIENT MORPHOLOGIQUE.....	51
III.3 APPROCHE REGION.....	51
III.3.1 SEGMENTATION PAR SEUILLAGE.....	52
III.3.2 TRAITEMENT A MULTIRESOLUTION.....	52
III.3.2.1 DEFINITION DE LA PYRAMIDE.....	53
III.3.2.2 PYRAMIDE GAUSSIENNE.....	53
III.4 CONCLUSION.....	54
CHAPITRE IV.....	55
IV.1 INTRODUCTION.....	55
IV.2 DEFINITION.....	55
IV.3 LES ALGORITHMES DE SQUELETTISATION.....	56
IV.3.1 LES ALGORITHMES A CRITERE TOPOLOGIQUE.....	56
IV.3.1.1 L’algorithme de Zhang et Suen.....	57
IV.3.1.2 L’algorithme de Serra.....	59
IV.3.1.3 L’algorithme de Rosenfeld.....	60
IV.3.2 LES ALGORITHMES A CRITERES METRIQUES.....	60
IV.3.2.1 L’algorithme de Marthon.....	61
IV.4 BINARISATION.....	63
IV.4 APPLICATION.....	65
IV.5 CONCLUSION.....	66
CHAPITRE V.....	69
V.1 INTRODUCTION.....	69
V.2 DEFINITIONS.....	70
V.2.1 PARCOURS D’ARBRE.....	70
V.2.2 DEFINITIONS ET NOTATIONS.....	73
V.3 PRINCIPE DE L’ETIQUETAGE.....	75
V.4 PARCOURS DE L’ARBRE.....	75
V.5 CODE DE FREEMAN.....	76
V.7 CONCLUSION.....	79
CHAPITRE VI.....	82
VI.1 INTRODUCTION.....	82
VI.2 ETUDE DE LA FAMILLE C6000.....	82
VI.2.1 ARCHITECTURE DU C6000.....	82
VI.2.2 LE CPU.....	84
VI-2-3 LES TIMERS.....	85
VI .2.4 LES REGISTRES.....	85
VI-2-5 LES MsBSP.....	86

VI.3 OUTILS DE DEVELOPPEMENT	87
VI.3.1 CODE COMPOSER STUDIO.....	87
VI.3.1.1 GENERATION DE CODES	88
VI.3.2 CARTE DSK.....	93
VI.3.3 LE RTDX.....	93
VI.4 CONCLUSION	95
VII.I INTRODUCTION.....	99
VII.2 CONFIGURATION DU CCS	99
VII.3 GESTION D'UN PROJET	101
VII.3.1 CREATION D'UN NOUVEAU PROJET.....	101
VII.3.2 AJOUT DE FICHIERS AU PROJET	103
VII.3.3 EXECUTION D'UN PROGRAMME.....	104
VII.3.4 CHARGEMENT ET SAUVEGARDE DES DONNEES	104
VII.4 SIMULATION.....	107
VII.4.1 FILTRAGE.....	107
VII.4.2 DETECTEURS DE CONTOURS	108
VII.4.3 BINARISATION.....	109
VII.4.4 SQUELLETISATION.....	110
VII.4.5 L'ETIQUETAGE	114
VII.5 CONCLUSION.....	114
CONCLUSION GENERALE	115
PERSPECTIVE.....	116

Liste des Figures

Figure I.1- Anatomie générale du cœur humain.....	6
Figure I.2- Division du cœur en 4 cavités.....	7
Figure I.3- Les trois dominances possibles du réseau coronaire.	8
Figure I.4- Anatomie générale des arbres coronaires droit et gauche.	9
Figure I.5- Cathéterisation.....	11
Figure I.6- Cathéter et injection du produit de contraste.....	11
Figure I.7- Structures parasites dans les angiogrammes.....	12
Figure I.8- Effets des distorsions en S (à gauche) et en coussin (à droite).....	15
Figure I.9- Paramètres d'acquisition CRA/CAU, LAO/RAO, et SID.....	17
Figure I.10- Représentation schématique du mode d'acquisition rotationnel et des différents points de vue qu'il propose au sein d'une même séquence.....	18
Figure II.1- Histogramme (b) des niveaux de gris de l'image (a).....	24
Figure II.2- bruit lié à la nature de la scène, apparition indésirable d'objets sur la scène colonne vertébrale et thorax.....	28
Figure II.3- Deux voisinages du point : (a) 8-connexité, (b) 4-connexité.....	29
FigureII.4- A droite le négatif de l'image originale.....	30
Figure II.5- Binarisation de l'image, seuil choisi expérimentalement.....	31
Figure II.6- Modification d'histogramme.....	31
Figure II.7- (a)Image d'angiographie initiale et (b) image traitée par transformation linéaire.....	32
Figure II.8- Application du filtre moyenneur 3x3.....	34
Figure II.9- application du filtre smooth.....	35
Figure II.10- Domaines de Nagao.....	37
Figure II.11- Image érodée après inversement des niveaux de gris.....	38
FigureII.12- Image dilatée après inversement des niveaux de gris.....	39
Figure II.13- Ouverture de l'image après inversement des niveaux de gris.....	39
Figure II.14- Fermeture de l'image après inversement des niveaux de gris.....	40
Figure II.15- Filtrage homomorphique.....	40
Figure III.1- Opérateur de Robert (à droite) appliqué à l'image originale.....	46
Figure III.2- Application des opérateurs de Prewitt (a) et de Sobel (b) à l'image originale....	47
Figure III.3- Application de l'opérateur MDIF sur l'image originale.....	48

Figure III.4- application de l'opérateur de Kirsh.....	49
Figure III.5- application du gradient morphologique après un lissage par le filtre moyennneur.....	50
Figure III.6- Diminution de la résolution d'un facteur de 2 de droite à gauche (128x128–64x64- 32x 32).....	53
Figure IV.1- Exemples de squelettes pour des formes simples.....	55
Figure IV. 2- Evolution du front enflammé dans une forme.....	56
Figure IV.3- Résultats de la binarisation pour différents seuils.....	63
Figure IV.4- Amélioration de la binarisation.....	65
Figure IV.5- Squelettes obtenu par différents algorithmes.....	66
Figure V.1- Exemple d'arbre.....	70
Figure V.2- Parcours de l'arbre.....	71
Figure V.3- Parcours préfixé de l'arbre.....	71
Figure V.4- Etiquetage de l'arbre dans l'ordre préfixé.....	72
Figure V.5- Etiquetage de l'arbre dans l'ordre ostfixé.....	72
Figure V.6- Codage de Freeman.....	76
Figure V.7- Résultats de l'étiquetage.....	78
Figure VI.1 : Block Diagram du TMS320C6211.....	83
FigureVI.2 : Bus internes	85
FigureVI.3 Périphériques internes.....	86
FigureVI.4- Build Options.....	89
FigureVI.5- Processus du développement du code exécutable.....	90
FigureVI.6- Création d'un nouveau projet.....	92
Figure VI.7- Différent fichiers composant un projet.....	93
FigureVI.8- Carte DSK.....	94
FigureVI.9- Connexion entre hôte et cible via JTAG.....	95
FigureVI.11 Etablissement de la connexion PC-Cible.....	96
FigureVI.12 : Connexion Cible-PC.....	97

INTRODUCTION GENERALE

Les maladies cardio-vasculaires désignent l'ensemble des pathologies touchant l'appareil circulatoire, qui comprend le cœur, le réseau artériel et le réseau veineux. Les facteurs de risque de ces maladies sont bien caractérisés : hérédité, régime alimentaire, tabagisme, alcoolisme, hypertension, et mode de vie sédentaire.

Actuellement, dans les pays développés, les maladies cardio-vasculaires sont la première cause de mortalité. D'après l'OMS, elles seront aussi la première cause de mortalité à l'échelle mondiale dès 2008. Elles représentent donc un enjeu de santé majeur pour l'ensemble des populations humaines.

Le domaine dans lequel se situe ce travail est celui de la segmentation d'images, ou de manière plus générale, celui de l'interprétation d'images angiographiques appliquées au réseau artériel coronaire.

Ce projet de fin d'études rentre dans le cadre des travaux de recherche du Dr.L.HAMAMI, il vise la segmentation par étiquetage des artères coronaires ainsi que la faisabilité de l'implémentation de ces algorithmes sur le processeur C6211 de la famille C6000 de Texas Instruments, afin de réaliser dans l'avenir un système de traitement d'images autonome ou embarqué. La partie théorique sera conclue par une application et création d'une librairie de fonctions pouvant être utilisée sous Code Composer Studio.

Ce rapport comporte les chapitres suivants :

Le chapitre I explique brièvement le fonctionnement du cœur, le rôle des artères coronaires dans l'irrigation de ce dernier, ainsi qu'un état de l'art de l'angiographie avec explication de son rôle dans la détection des éventuelles pathologies cardiovasculaires (sténose ou thrombose).

Le chapitre II donne quelques notions sur le traitement d'images en général, et les algorithmes de prétraitement nécessaires pour améliorer la qualité de l'image.

Le chapitre III est consacré à la segmentation, donc aux algorithmes de détection de contours adoptant les approches dérivatives de premier ordre et du deuxième ordre, ainsi que l'approche région.

Le chapitre IV présente la squelettisation, phase primordiale pour l'étiquetage des artères coronaires. Des algorithmes sont détaillés comme celui de Marthon ou Zang et Suen.

Dans le chapitre V on présentera l'algorithme d'étiquetage avec quelques rappels théoriques.

Le chapitre VI, est une étude de la famille C6000 en particulier le processeur C6211 mis en œuvre sur une carte DSK (support de développement), ainsi qu'une présentation des outils de développement logiciels et matériels employés dans la réalisation de l'application en particulier le Code Composer Studio de Texas Instruments.

Dans le dernier chapitre (VII), les algorithmes présentés aux chapitres précédents feront l'objet de simulation sous CCS. L'algorithme d'étiquetage sera détaillé afin d'expliquer le fonctionnement d'un programme sous CCS, nous donnerons ainsi les différents résultats obtenus

CHAPITRE I

**ANATOMIE DU CŒUR ET
ACQUISITION DES
IMAGES**

CHAPITRE I

ANATOMIE DU CŒUR ET ACQUISITION D'IMAGES

I.1 INTRODUCTION

Dans ce chapitre nous allons présenter l'anatomie générale du cœur humain et de son réseau artériel, vu que notre travail portera sur le traitement des images cardiaques afin de détecter d'éventuelles pathologies touchant notamment l'arbre coronaire (sténoses, thromboses). Nous tenterons d'expliquer l'une des modalités les plus utilisées pour acquérir ces images : l'angiographie à rayons X.

I.2 ANATOMIE DU CŒUR, ET DES ARTERES CORONAIRES

Le cœur est l'organe principal de l'appareil circulatoire humain. Il est la pompe qui assure la circulation du sang à travers le corps, il apporte aux organes un sang riche en oxygène et évacue le sang altéré par le dioxyde de carbone, produit par la respiration des cellules.

C'est un muscle creux qui comprend deux oreillettes et deux ventricules. Ces quatre cavités sont séparées par des cloisons s'exprimant extérieurement par des sillons. On décrit les sillons inter-ventriculaires (entre les deux ventricules), les sillons inter-auriculaires (entre les deux oreillettes), les sillons auriculo-ventriculaires (entre oreillettes et ventricules).

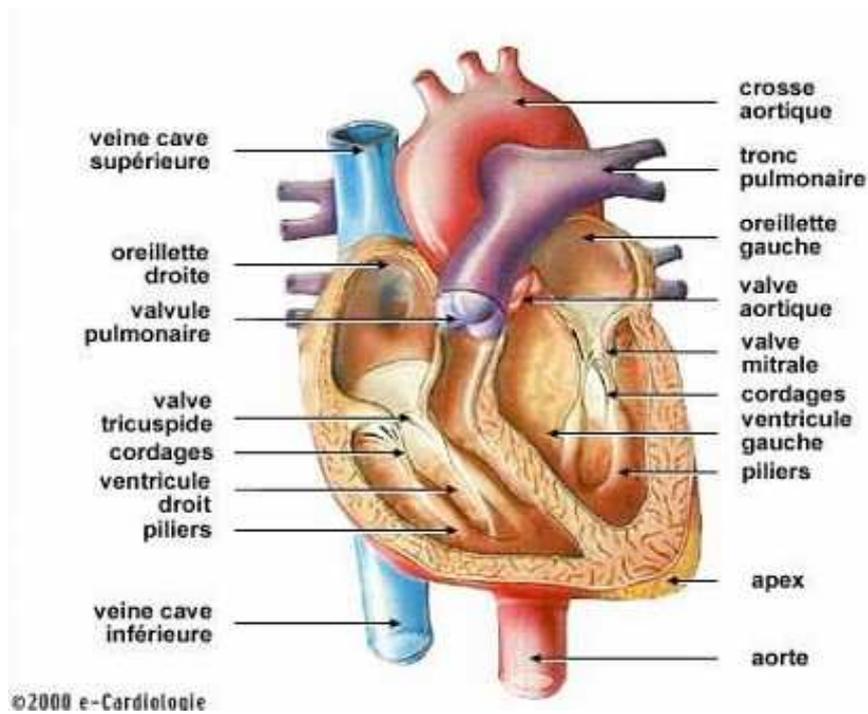


Figure I.1- Anatomie générale du cœur humain

Le cœur est entouré, à sa surface d'artères et de veines, selon des trajets en «couronne» autour du muscle cardiaque, d'où leur nom d'artère «coronaires» (latin corona, couronne). Après un cheminement de quelques centimètres à la surface du cœur, les branches de division des gros troncs s'enfoncent dans le muscle cardiaque. Le diamètre des artères coronaires est de plus en plus faible de la proximalité à la distalité.

Le muscle cardiaque s'appelle myocarde (grec mus, muscle et kardia, cœur). Le myocarde est doublé en dedans de l'endocarde (grec endon, dedans) et en dehors du péricarde (grec péri, autour).

Le ventricule gauche (left ventricle) éjecte le sang dans la grande circulation par l'aorte qui donne les artères à destinée périphérique. Le ventricule gauche a des parois épaisses, autorisant des régimes de pression élevés. L'éjection ventriculaire se fait vers l'aorte grâce à l'étanchéité de la valve mitrale qui permet l'admission de sang dans le ventricule mais pas son reflux vers l'oreillette. De même une fois éjecté dans l'aorte, le sang ne reflue pas dans le ventricule gauche grâce à l'étanchéité des valves aortiques sigmoïdes.

Les artères coronaires sont au nombre de deux: la gauche et la droite. Elles naissent de l'aorte initiale, au-dessus des cusps aortiques (renflements de l'aorte à son origine) et des valvules aortiques. Leur origine est appelée ostium (orifice en latin).

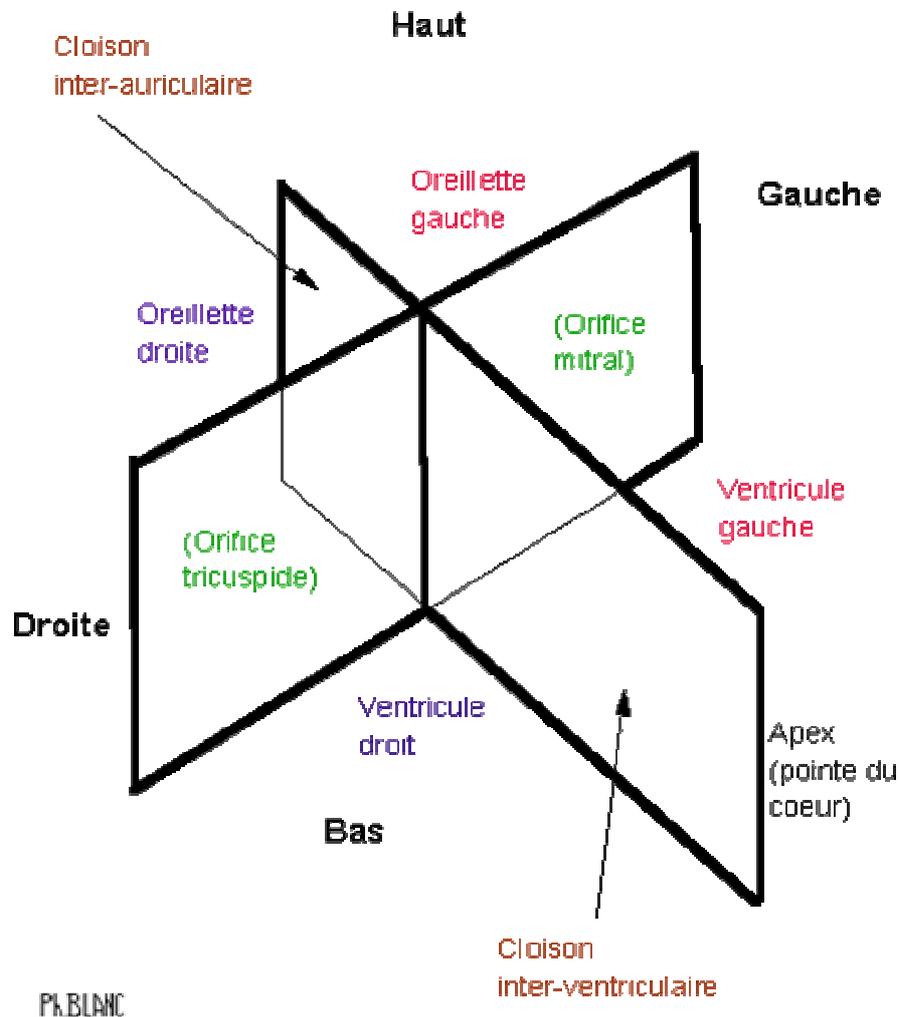


Figure I.2- Division du cœur en 4 cavités.

La variabilité anatomique inter-individu de l'arbre coronaire est très importante. Les artères coronaires naissent à la base de l'artère aorte au niveau d'orifices appelés *ostium coronaire gauche* (le plus souvent unique) et *ostium coronaire droit* (parfois multiple). L'arbre coronaire est donc divisé en deux branches : la coronaire droite et la coronaire gauche. L'importance relative de ces deux branches est variable selon les personnes. Cette importance relative est décrite en terme de réseau coronaire à dominance droite, à dominance gauche, ou équilibré. La dominance droite est prévalente avec typiquement 70% de la population, contre 20% pour la dominance gauche, et 10% pour les réseaux équilibrés. [1]

La figure 1.3 montre les aspects généraux des trois types de dominance.

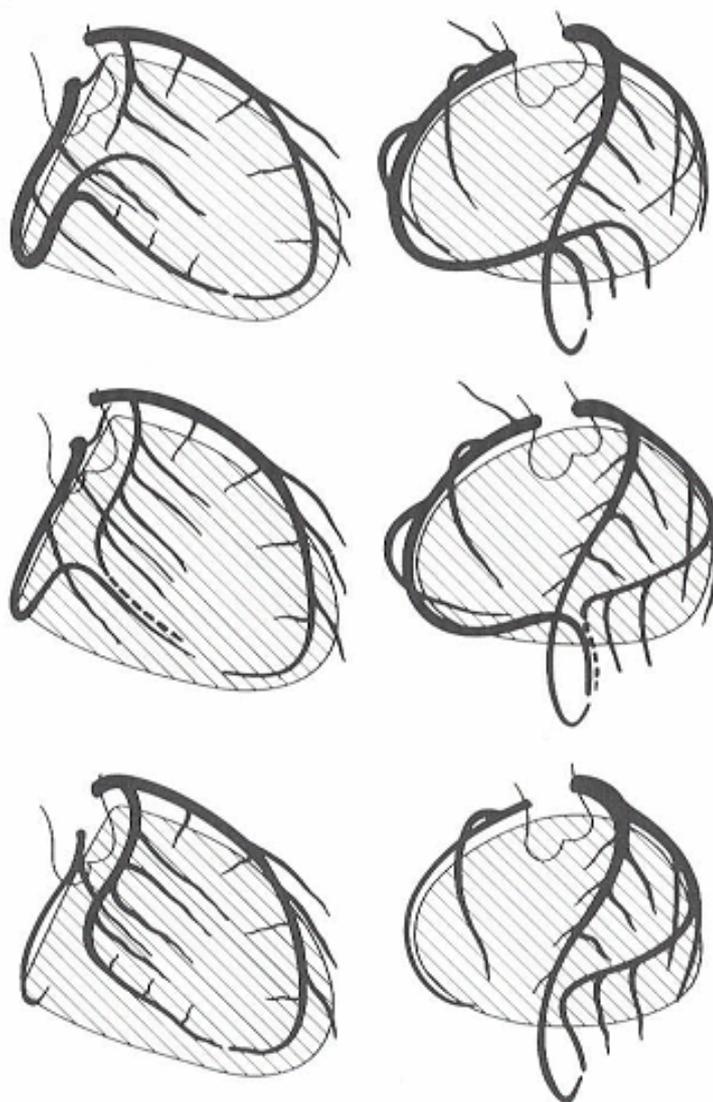


Figure I.3- Les trois dominances possibles du réseau coronaire. De haut en bas, représentation schématique de réseaux coronaires à dominance droite, équilibrée, et gauche. Ce sont, à gauche, des vues antérieures et, à droite, des vues latérales droites.

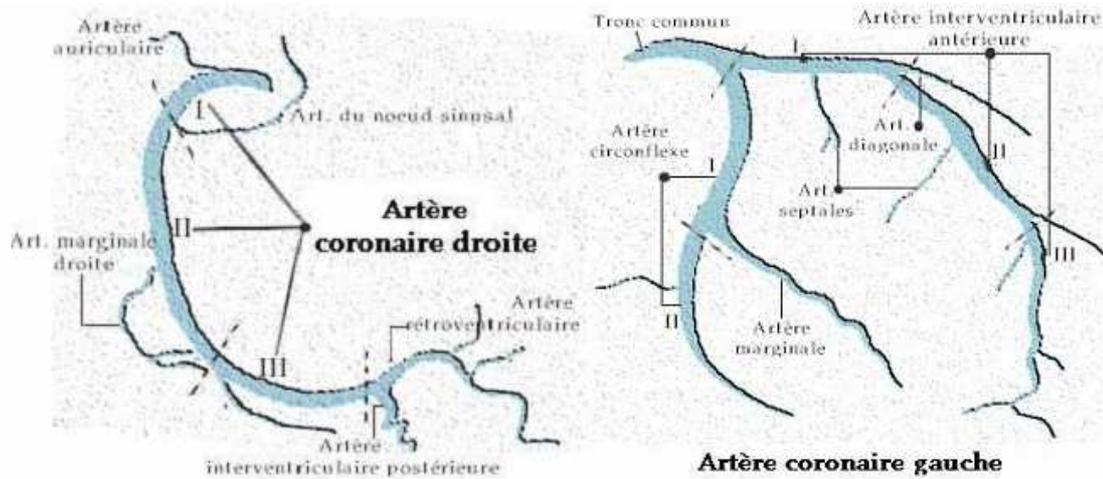


Figure I.4- Anatomie générale des arbres coronaaires droit et gauche. Les deux branches principales se ramifient progressivement pour assurer une micro-circulation homogène dans l'ensemble du myocarde.

Comme illustré dans la figure 1.4, l'artère coronaire droite donne naissance successivement aux artères marginale droite, inter-ventriculaire postérieure et rétro-ventriculaire gauche.

L'artère coronaire gauche donne naissance à l'artère inter-ventriculaire antérieure (de laquelle partent les artères diagonales et septales), puis à l'artère circonflexe (de laquelle partent les artères marginales gauches, voire l'artère inter-ventriculaire postérieure dans le cas d'une dominance gauche). Alors que les branches principales ont un trajet de surface ou épicaordique, les branches ramifiées ont parfois des parcours au sein même du muscle, ou parcours intra-pariétal.

Au niveau du tronc coronaire, le diamètre typique des artères est de 5 millimètres. Les deux branches principales se subdivisent rapidement. Les branches principales ont des gabarits diminuant progressivement de 4 à 2 millimètres. Au fur et à mesure des ramifications, les artères coronaaires atteignent rapidement des diamètres inférieurs à 1 millimètre, jusqu'à devenir des artérioles puis des capillaires.

I.3 ACQUISITION DES IMAGES

I.3.1 L'angiographie par rayons X

I.3.1.1 Description

L'angiographie par rayons X est une modalité d'imagerie médicale essentiellement anatomique. Elle permet l'observation des vaisseaux (angio vient du grec *aggeion* qui signifie vaisseau) depuis l'extérieur du corps grâce à la conjonction de deux propriétés physiques [2,3] :

- la radio-transparence partielle du corps humain aux rayons X,
- la radio-opacité importante d'un produit injectable dans les vaisseaux.

De manière analogue à une radiographie pulmonaire classique, une source émet un faisceau de rayons X vers un détecteur, qui peut être argentique, analogique ou numérique. La source et le détecteur sont positionnés de manière à ce que l'objet que l'on cherche à observer en transparence soit situé sur le chemin des rayons X.

Schématiquement, les rayons X traversent les matériaux de faible densité et sont absorbés par les matériaux de forte densité. Ainsi, la plupart des tissus du corps humain, à l'exception des os, laissent passer en grande partie les rayons X. Sur le chemin le menant de la source au détecteur, plus un faisceau de rayons X rencontre de structures denses, plus il sera atténué et plus faible sera le signal recueilli sur le détecteur. C'est pour cela, par exemple, que sur les négatifs de radiographies pulmonaires, les structures osseuses, plus denses, apparaissent claires alors que les tissus mous apparaissent sombres.

L'information que présentent les angiographies est de nature anatomique car elle est liée à la densité des milieux traversés, ce qui dans notre cas permet d'observer la géométrie des artères, projetée sur le détecteur.

Comme le but de l'angiographie est de visualiser les structures vasculaires et que celles-ci ne sont pas naturellement radio-opaques, on injecte dans les vaisseaux à examiner un produit dit de contraste, contenant un composé lourd : l'iode. Dans le cas de l'angiographie des artères coronaires, le produit de contraste est amené à l'entrée du tronc coronaire par un tube souple, fin et creux, appelé cathéter, introduit dans le corps par l'artère fémorale ou par l'artère humérale, comme illustré dans la figure 1.5.

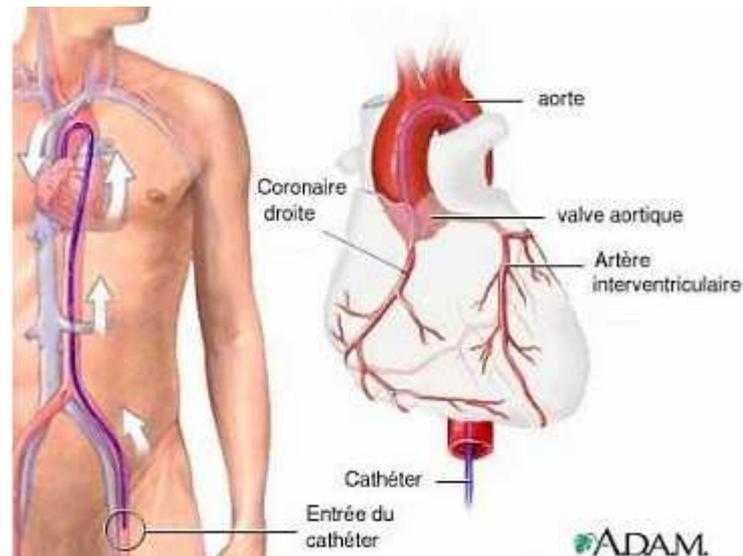


Figure I.5- Cathéterisation.

La figure 1.6 montre un angiogramme dans lequel le cathéter est visible avant l'injection du produit de contraste, et deux autres angiogrammes montrant l'injection du produit de contraste et sa progression rapide dans les vaisseaux, liée au flux sanguin. Outre les vaisseaux injectés, on peut observer d'autres structures denses : les os, tels les côtes et les vertèbres, les tissus épais, denses, ou fibreux, tels les bronches, ou le diaphragme et certains objets artificiels, tels les sutures, le cathéter, et les stents. Comme illustré dans la figure 1.7, tous ces objets présents dans les angiographies seront autant de structures parasites pouvant perturber l'imagerie des structures d'intérêt, c'est-à-dire des artères coronaires.

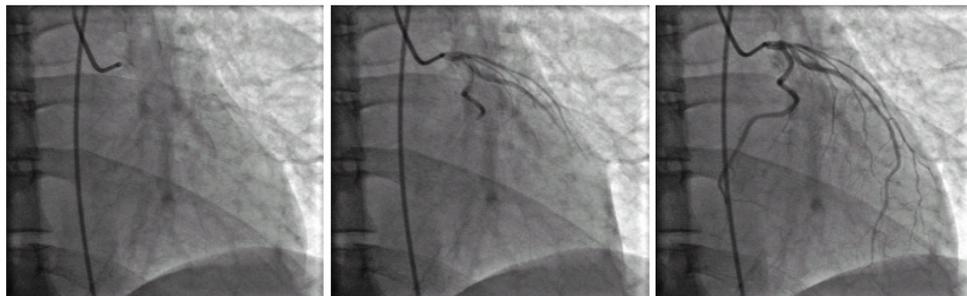


Figure 1.6- Cathéter et injection du produit de contraste.

De gauche à droite : angiogramme dans lequel le cathéter amené par voie fémorale est visible mais les vaisseaux sont invisibles car non encore injectés de produit de contraste, angiogramme dans lequel l'injection commence et dévoile la partie proximale des artères coronaires, et angiogramme dans lequel l'injection du réseau coronaire est complète et permet d'observer l'ensemble des vaisseaux.

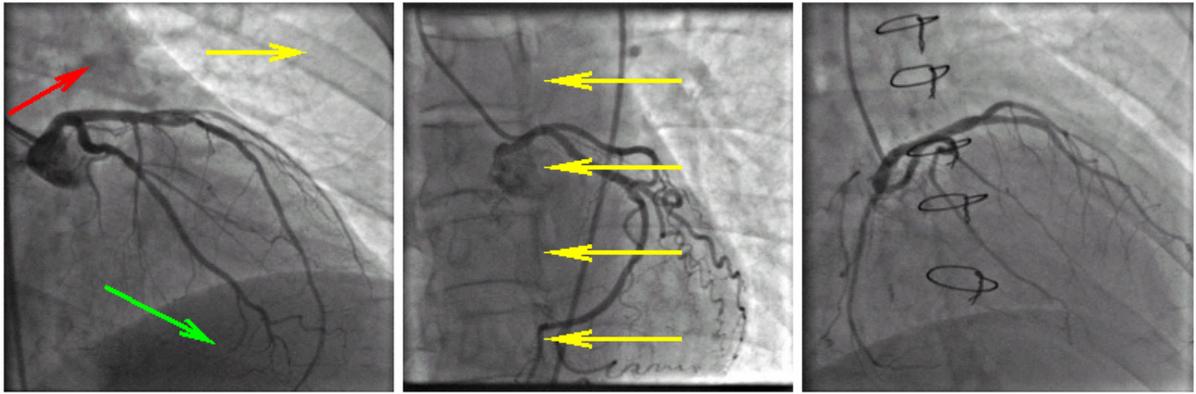


Figure I.7- Structures parasites dans les angiogrammes. De gauche à droite : angiogramme montrant des côtes (en jaune), le diaphragme (en vert), des structures pulmonaires denses (en rouge), angiogramme montrant quatre vertèbres, et angiogramme montrant des sutures. Ces structures perturbent l'observation des artères coronaires.

I.3.1.2 Formation de l'image

L'image angiographique est le produit de trois processus : l'émission, la transmission, et la détection de rayons X.

I.3.1.3 l'émission de rayons X

L'émission de rayons X se fait à partir d'un tube sous vide. Un filament, constituant la cathode, est chauffé et émet des électrons par effet thermoélectronique. Une haute tension appliquée entre cette cathode et une anode accélère et attire les électrons vers l'anode. Ceux-ci percutent l'anode avec une haute énergie, engendrant des changements d'orbites électroniques et, par effet photo-électrique, un rayonnement dont les longueurs d'onde sont caractéristiques du matériau de l'anode (en général molybdène ou rhodium) et choisies dans le domaine rayons X du spectre lumineux.

Un système de filtre (en général en aluminium) arrête les radiations parasites de basse énergie et un système de collimation ne laisse s'échapper du tube que les rayons qui ont la direction voulue.

I.3.1.4 Transmission de rayons X

L'interaction entre rayons X et matière est un processus extrêmement complexe dont nous ne retiendrons que les deux effets majeurs :

- l'absorption,
- la diffusion.

L'absorption, due à l'effet photo-électrique, se traduit par la perte d'énergie progressive des rayonnements X qui entrent en collision avec des électrons d'orbites basses. Plus un matériau est dense, plus il va atténuer l'intensité d'un faisceau de rayons X. L'effet de l'atténuation est une retenue plus ou moins importante des rayons X traversant un matériau. La proportion de rayons X qui traversent le matériau est caractéristique de sa densité. La diffusion, liée à l'effet Compton, va peu diminuer l'énergie du rayonnement X, mais va en modifier la direction. L'effet de cette interaction est une diffusion des rayons autour de leur direction d'origine. Un faisceau unidirectionnel devient un faisceau en forme de tromblon sous l'effet de la diffusion. Il induit donc un flou sur la direction du rayonnement X. Par la modification de la direction du rayonnement initial, la diffusion des rayons X induit aussi l'atténuation du rayonnement direct. En angiographie, on cherche à observer l'effet de l'absorption et à limiter l'effet de la diffusion. Le rayonnement diffusé peut être réduit par l'emploi de filtres sélectifs selon l'énergie du rayonnement, et de collimateurs, sélectifs selon la direction du rayonnement. En effet, ces deux caractéristiques sont modifiées par rapport au rayonnement initial. Nous introduisons maintenant l'équation de Lambert-Beer qui donne la loi d'absorption des rayons X. Soit $I(x)$ l'intensité d'un faisceau à la position x , soit dI la variation d'intensité du faisceau sur un élément de longueur dx , alors nous avons la relation :

$$dI = -\mu(x).dx \quad (1.1)$$

Le coefficient $\mu(x)$ est appelé coefficient d'absorption linéaire locale en x . En pratique, ce coefficient dépend, d'une part, de l'énergie du faisceau incident et, d'autre part, du milieu traversé, et en particulier de sa densité. En intégrant la relation élémentaire (1.1), nous obtenons l'équation de Lambert-Beer, qui caractérise l'intensité d'un faisceau, après qu'il ait traversé une épaisseur y de matériau, dont le support est supposé commencer en $x = 0$:

$$I(y) = I(0) \cdot e^{-\int_0^y \mu(x) dx} \quad (1.2)$$

L'interprétation de ce phénomène de transmission des rayons X est la suivante : l'intensité du faisceau résiduel à la sortie du matériau absorbant est (à l'application d'un logarithme près) l'intégrale sur le rayon des absorptions linéaires locales traversées.

I.3.1.5 Détection de rayons X

La détection des rayons X qui ont traversé le matériau à observer se fait sur un capteur qui comptabilise les photons X. Nous décrivons maintenant les trois évolutions technologiques majeures dans la détection des photons X, dans l'ordre chronologique de leur introduction :

- la plaque au phosphore,
- l'intensificateur,
- le panneau plat numérique.

a) Plaque au phosphore A l'image d'un appareil photographique, le détecteur le plus ancien en radiologie conventionnelle est la plaque, qui n'est alors pas argentique, mais dite au phosphore. Celle-ci va réagir d'autant plus qu'elle reçoit un rayonnement X important. Le développement des plaques, via un négatif, produit des clichés semblables à des photographies, mais pour lesquelles la lumière observée n'est pas celle du spectre visible mais celle des rayons X. Ces plaques sont maintenant délaissées (à l'exception de la radiologie conventionnelle) au profit de technologies offrant un meilleur contraste dans les images.

b) Intensificateur Afin d'augmenter la dynamique du détecteur, et donc le contraste dans les images, on utilise un dispositif électronique dit intensificateur. Le processus d'intensification est le suivant : chaque photon X incident est converti en électron par effet photoélectrique, le signal électronique est intensifié par un dispositif électrique de gain, puis converti en photons lumineux qui sont alors filmés. Le défaut principal de ce détecteur est l'introduction de distorsion dans l'image. Il existe deux types distincts de distorsion. La première, dite en S, est liée à la sensibilité de la trajectoire des électrons aux champs magnétiques environnants, et en particulier au champ magnétique terrestre. La seconde, dite en coussin, est due à la courbure du détecteur qui provoque une déformation radiale de l'image. La figure 1.8 illustre ces deux types de distorsion dans les images. Les phénomènes de distorsion et une trop faible dynamique de l'image dans certaines conditions ont conduit les constructeurs à introduire des détecteurs plans numériques.

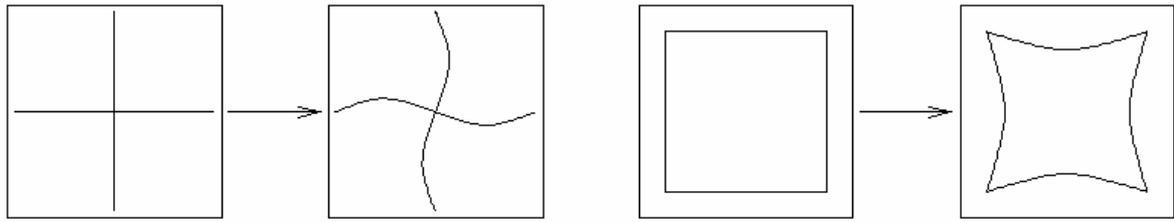


Figure 1.8- Effets des distorsions en S (à gauche) et en coussin (à droite).

c) **Panneau plat numérique** L'introduction récente de détecteur à base de panneau plat numérique a considérablement accru la qualité des images acquises. Dans le cas du panneau plat numérique, les photons X sont convertis en photons lumineux par un scintillateur en iodure de césium cristallin, une photodiode en silicium amorphe convertit les photons lumineux en électrons qui sont ensuite comptabilisés par une électronique à bas bruit. Cette méthode de détection s'apparente à celle utilisée dans un capteur CCD.

Les images ne souffrent plus de distorsions en S et en coussin (leurs causes ayant été supprimées) et sont moins bruitées, grâce à la réduction du nombre d'étapes intermédiaires de conversion.

Enfin, la plus grande dynamique du détecteur permet d'observer des changements importants des intensités détectées. Cette caractéristique est importante dans le cas d'une acquisition rotationnelle, où l'épaisseur projetée du patient varie considérablement (dans un rapport de 1 à 3) selon le point de vue. Par exemple le modèle d'angiographie Innova 2000 utilise la technologie du panneau plat numérique pour la détection des rayons X. Nous donnons quelques valeurs caractéristiques des images acquises par ce système. La résolution spatiale des images est de 0,2 millimètres, les tailles des champs de vue utilisés pour les artères coronaires sont 17 centimètres et 20 centimètres. Les tailles des images associées à ces deux tailles de champ sont respectivement 864 et 1000 pixels. En sortie de la chaîne de traitement, chaque pixel est codé sur 8 bits.

I.3.1.6 Trajectoires d'acquisition et séquences d'images

Nous avons décrit le mode de formation d'une image. En routine clinique, les images sont acquises en rafale, sous forme de séquences. La fréquence d'acquisition est 30 Hz. Sur un angiographe de type LC, une vue est caractérisée par trois valeurs qui donnent la configuration géométrique de l'arceau (le L est le plus souvent conservé fixe). Ces trois valeurs sont :

- un angle dans le plan sagittal, nommé cranial /caudal et noté CRA/CAU,
- un angle dans le plan axial, nommé antérieur oblique droit/antérieur oblique gauche et noté LAO/RAO,
- une distance, entre émetteur et détecteur de rayons X, notée SID.

La figure 1.9 illustre ces trois paramètres géométriques d'acquisition.

I.3.1.7 Mode statique

Le protocole classique d'angiographie coronaire consiste à conserver l'ensemble du système dans une position statique et à acquérir une séquence d'images durant 2 à 5 secondes. Dans ce cas, les angles CRA/CAU et LAO/RAO et la distance SID restent constants durant l'acquisition.

I.3.1.8 Mode rotationnel

Un mode d'acquisition introduit récemment en imagerie à rayons X des artères coronaires est le mode d'acquisition rotationnel, nommé cardiac spin [2].

Dans sa forme actuelle, il consiste en une rotation de l'arceau C dans le plan axial, avec une valeur d'angle CRA/CAU constante, égale à 0° . L'angle LAO/RAO varie quant à lui de -100° à 100° , pour les rotations les plus amples. La distance SID reste bien entendu constante au cours de cette rotation. La vitesse de rotation maximale du système est de 40° par seconde, ce qui donne un temps total d'acquisition de 3 à 5 secondes selon l'amplitude de la rotation.

La figure 1.10 illustre ce mode d'acquisition rotationnel et les différents points de vue qu'il permet d'acquérir dans une même séquence.

A chaque image, et donc à chaque valeur d'angle LAO/RAO, d'angle CRA/CAU et de distance SID fixés, on peut associer une application qui à un point de l'espace tridimensionnel associe sa position projetée dans l'image. On appelle cette application matrice de projection. Dans le cas d'une séquence d'images acquises en rotation, on dispose donc aussi de la séquence des matrices de projection associées.

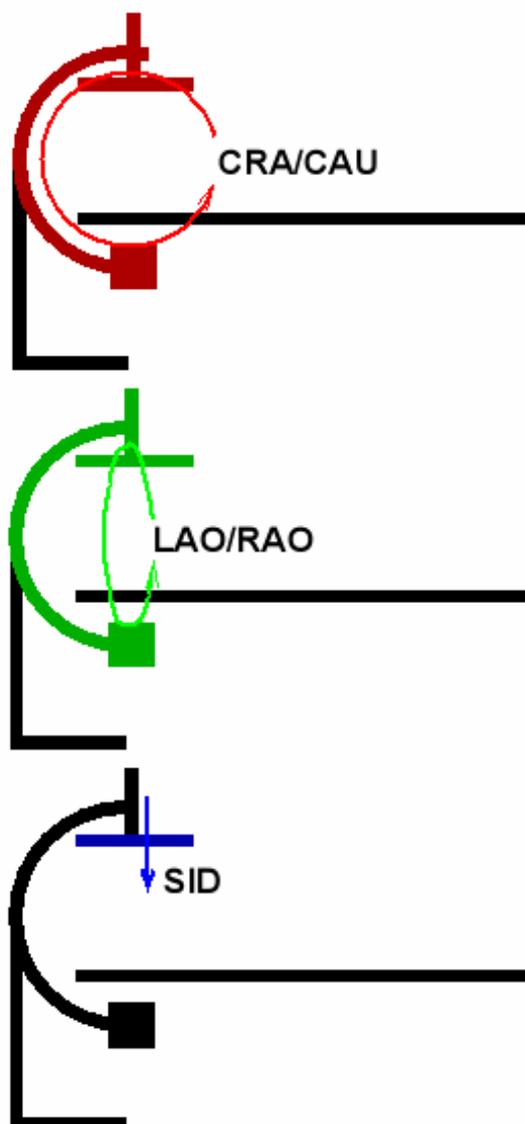


Figure 1.9- Paramètres d'acquisition CRA/CAU, LAO/RAO, et SID



Figure I.10- Représentation schématique du mode d'acquisition rotationnel et des différents points de vue qu'il propose au sein d'une même séquence.

1.4 CONCLUSION

Dans ce chapitre nous avons fait une première approche de l'imagerie médicale en traitant brièvement l'angiographie à rayons X, ainsi que l'anatomie du cœur et de son réseau artériel. D'autres modalités d'acquisition d'images cardiaques ont été exposées en annexe.

Dans le chapitre suivant, nous allons faire une introduction au monde du traitement du traitement d'images numériques, avec les différentes méthodes d'amélioration et de filtrage des images numériques.

CHAPITRE II

GENERALITES ET

NOTIONS

FONDAMENTALES

Chapitre II

GENERALITES ET NOTIONS FONDAMENTALES

II.1 INTRODUCTION

Le traitement d'images est l'ensemble des méthodes qui cherche à rendre une image plus intelligible ou manipulable, soit pour l'être humain ou par la machine, en mettant en claire certaines de ces caractéristiques, en vue d'une application donnée, comme déceler une sténose ou une thrombose dans une artère, en imagerie médicale, ou rendre plus compréhensible une image planétaire, d'où la diversité des traitements qui peuvent être appliqués. Une caractéristique très importante et en même temps très enrichissante du traitement d'image est son caractère pluridisciplinaire. Il convient de préciser qu'il n'y a pas de méthodes ou algorithmes prédéfinis qui peuvent être appliqué partout.

Dans ce chapitre nous allons essayer de faire un survol sur les principes de base en traitement des images numériques. On commencera par des définitions, puis on donnera quelques méthodes de prétraitement, comme l'amélioration de contraste, filtrage,...etc.

II.2 GENERALITE

II.2.1 Image numérique

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 de couleurs prélevées à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont rangées sous forme de matrice de n lignes et p colonnes [9].

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,P-1) \\ f(1,0) & f(1,1) & & & f(1,P-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(N-1,0) & \dots & & & f(N-1,P-1) \end{pmatrix}$$

II.2.2 L'acquisition de l'image

Différents types de capteurs sont disponibles pour générer des images. Ils se distinguent par leur principe d'acquisition, leur vitesse d'acquisition, leur résolution spatiale, leur gamme spectrale ou encore leur dynamique. Néanmoins, les capteurs utilisés le plus fréquemment dans les systèmes de vision industriels sont les circuits à transfert de charges CCD. Ils résultent de l'association d'une cellule photosensible et d'un dispositif de transfert de charge. Notons que ces capteurs réalisent l'échantillonnage de l'image, mais pas sa quantification [10].

La technologie CCD (Charged Couple Device) :

Un élément CCD est un semi-conducteur métal oxydes formé par une ligne de minuscules cellules photosensibles. Le nombre des cellules dans une ligne détermine la résolution dans le sens horizontal. Si une ligne contient, par exemple, 5000 cellules, un scanner A4 génère 5000 points sur les 21 centimètres de largeur. Si un décalage existe, même d'une seule cellule, des rayures apparaîtront certainement sur le scanner.

Nous distinguons deux familles de caméras CCD :

➤ **Les caméras matricielles :**

Les photodiodes sont assemblées en une matrice de n lignes et p colonnes.

➤ **Les caméras uni lignes :**

Comme leur nom l'indique, les photodiodes y sont agencées suivant une seule ligne. Elles ont l'avantage de posséder une résolution supérieure. Ce type de caméra est bien adapté à l'acquisition de documents, on déplace la feuille à vitesse constante perpendiculairement à la caméra et on fait une acquisition d'image suivant une période T [12].

II.2.3 Partitions élémentaires

Un pavage est une partition du plan, utilisant toujours les mêmes formes élémentaires. Les éléments constituant un pavage sont appelés tesselles. Ils existent trois types de pavage :

rectangulaire, hexagonal et triangulaire, de ces trois le pavage rectangulaire est le plus utilisé. La notion de récursivité du pavage est particulièrement importante pour les algorithmes de segmentation.

Un pavage est dit récursif si les tesselles peuvent être regroupés (resp. décomposés) en un ensemble de tesselles de taille plus petite (resp. plus grande).

Le pavage récursif présente l'intérêt de s'adapter à des analyses en multirésolution spatiale (changement d'échelle) [6].

II.2.4. Notion de pixel

Une image est constituée d'un ensemble de points appelés pixels (pixel est une abréviation de PICture ELement, c'est-à-dire qu'il s'agit du plus petit élément constitutif d'une image). C'est une entité calculable qui peut recevoir une structure et une quantification. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions. 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 (RVB), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

II.2.5 La résolution

La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouce)). Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors du processus de numérisation. Plus le nombre de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité d'information qui décrit cette structure est importante et plus la résolution est élevée.

II.2.6 Images 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 est donc représenté par un octet.

II.2.7 Histogramme

Soit p_i le nombre d'échantillons possédant la valeur du niveau i (i niveau de gris allant de 0 à 255), une fonction très utile en traitement d'image est la représentation des p_i en fonction de l'indice i , cette fonction s'appelle l'histogramme de l'image. C'est une fonction qui donne la fréquence d'apparition de tous les niveaux de gris dans une image donnée et qui ne tient pas compte de leurs distributions spatiales. Le calcul d'histogramme peut faire l'objet d'une diminution de l'erreur de quantification, une comparaison de deux images obtenues sous éclairages différents et une amélioration de certaines proportions afin d'extraire les informations utiles (figure 2.1) [5].

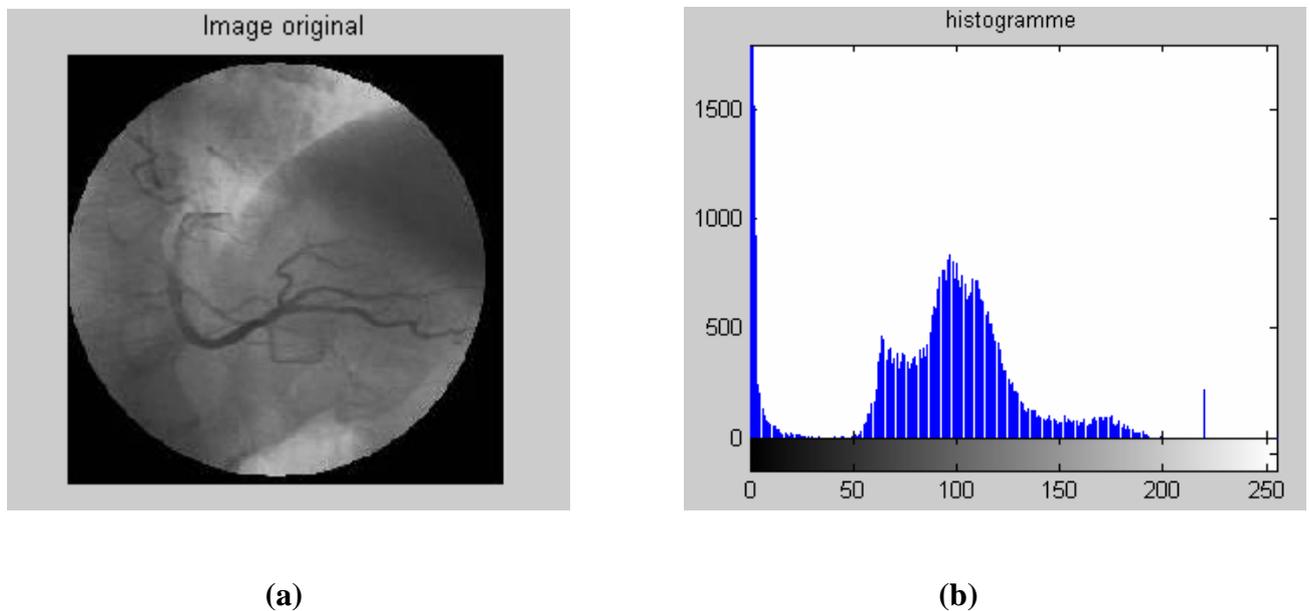


Figure 2.1- Histogramme (b) des niveaux de gris de l'image (a).

II.2.8 Images en couleurs

Il est parfois suffisant de représenter des images en noir et blanc, cependant 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 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,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle RVB) [14].

II.2.9 Intensité

L'intensité ou luminance est le caractère qui indique l'intensité de lumière perçue indépendamment de la couleur. Elle s'étend du noir au blanc avec toutes les nuances de gris si on ne voit pas la couleur.

II.2.10 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. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (2.1)$$

II.2.11 Le 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, de la qualité du cliché, du procédé de développement, du stockage, de la numérisation, etc....

Une caractéristique importante du bruit d'image est son comportement plus ou moins impulsionnel que l'on peut caractériser par la plus ou moins forte probabilité de valeurs aberrantes, ainsi que par son caractère ponctuel (pics parasites) ou continu.

II.2.12 Les sources de dégradations dans une image

Les techniques de restauration se proposent de réduire les effets des dégradations (ou bruits) subies par une image : il est donc utile, dans un premier temps, de voir quelles peuvent être les sources de dégradation. Il en existe quatre possibles :

- Bruit lié au contexte de l'acquisition.
- Bruit lié au capteur.
- Bruit lié à l'échantillonnage.
- Bruit lié à la nature de la scène.

II.2.12.1 Bruit lié au contexte de l'acquisition :

Dans cette première catégorie, on retrouve tous les événements inattendus, modifiant les conditions de l'acquisition du signal. L'exemple le plus simple est le bougé. On peut aussi imaginer une modification ponctuelle des conditions d'éclairage, conduisant à une sous ou sur illumination de l'objet observé. Le caractère inattendu de ces événements fait qu'il est pratiquement impossible d'en corriger les effets sans l'intervention d'un opérateur humain pour activer le bon processus de restauration (correction de flou, de bougé, de dérive lumineuse,...).

II.2.12.2 Bruit lié au capteur

Le capteur, s'il est de mauvaise qualité ou s'il est mal utilisé, peut introduire toute sorte de bruits. Nous pouvons ainsi obtenir une distorsion de la gamme des niveaux de gris, provoquant une saturation ou bien une distorsion géométrique de l'image équivalente à l'effet d'un miroir grossissant par exemple. Une mauvaise mise au point peut provoquer un flou. Bien entendu, la meilleure restauration possible consiste à agir directement au niveau du capteur.

Cependant, si cela n'est pas possible, la connaissance de l'appareil, permet le plus souvent, de construire un modèle a priori du phénomène de dégradation; une inversion systématique est alors envisageable. C'est le cas en microscopie électronique où il est très difficile d'obtenir des images de bonne qualité, mais où les phénomènes dégradants (produisant le plus souvent un mauvais contraste ou une dérive lumineuse) sont bien connus et modélisés.

II.2.12.3 Bruit lié à l'échantillonnage

Une image est une forme discrète. Il est donc nécessaire de passer du domaine continu au domaine discret, au niveau de la scène et au niveau de l'intensité lumineuse émanant de celle-ci. De manière générale, le spectre des intensités lumineuses noir et blanc est quantifié sur 256 niveaux de gris différents nécessitant 8 bits pour le stockage en chaque point.

Si le capteur est bien calibré, cette quantification ne provoque pas de bruit important. Il faut savoir que cette précision est de loin supérieure à ce que l'oeil humain utilise dans ses tâches quotidiennes. Par contre, la quantification de l'espace en $n \times n$ points est un inconvénient majeur; les capteurs actuels ne permettent pas d'obtenir une précision satisfaisante pour permettre un contrôle dimensionnel de formes.

De plus, tout objet dont la taille est inférieure à l'unité de tessellation [5] du plan disparaît ou bien n'est représenté que par un pixel. Cet effet est connu sous le nom d'effet "poivre et sel". Ce bruit est aussi généré par une texture dont les caractéristiques sont trop fines par rapport à la fréquence d'échantillonnage (phénomène de Moiré).

II.2.12.4 Bruit lié à la nature de la scène

L'environnement dans lequel se situe l'acquisition n'est pas toujours maîtrisable. En Télédétection, la couverture nuageuse dégrade la qualité des images satellites. De même, la présence des os au niveau de la cage thoracique perturbe le processus d'angiographie. Cependant, là encore, la connaissance a priori du phénomène perturbateur permet d'envisager une modélisation et donc une correction systématique.

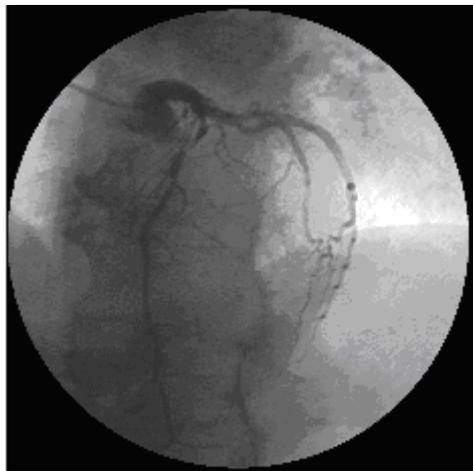


Figure 2.2- bruit lié à la nature de la scène, apparition indésirable d'objets sur la scène colonne vertébrale et thorax.

II.2.13 Voisinage et métriques associées

Les définitions de voisinage et des métriques associées sont liées au type de maillage, on ne parlera dans ce qui suit que du maillage carré.

Soit l'image discrète considérée comme un sous-ensemble borné de \mathbb{N}^2 . Chaque point P de l'image est un site dont les coordonnées $[i_p, j_p]$ sont des entiers ; $i_p \in \{1, \dots, M\}$ est l'indice de ligne et $j_p \in \{1, \dots, N\}$ est l'indice de colonne [5].

Les deux métriques couramment utilisées en maillage carré sont désignées par d_4 et d_8 .

La métrique d_4 (city block metric) est définie par :

$$d_4(P, Q) = |i_p - i_q| + |j_p - j_q|$$

Elle permet d'associer à P, un ensemble de points $v_4(p)$, appelé voisinage et défini par :

$$v_4 = \{Q \in S, d_4(P, Q) \leq 1\}$$

Ce voisinage est représenté sur la figure (2.2.a). Le point P possède 4 voisins adjacents situés à une distance $d_4 = 1$. On dit que v_4 est un voisinage 4-connexe.

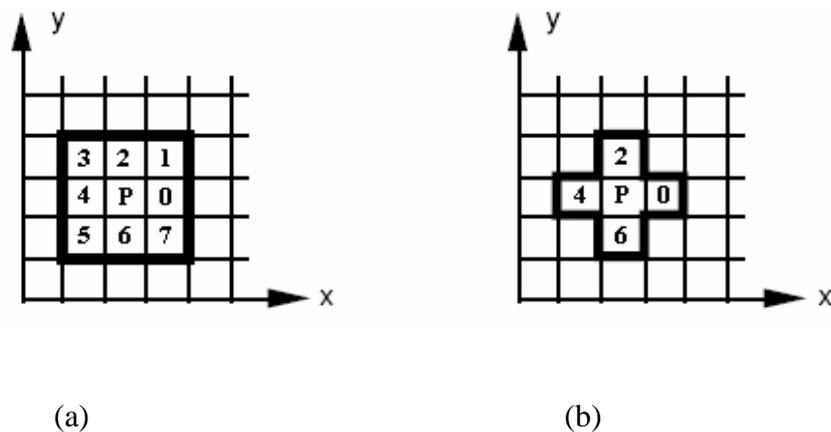


Figure 2.3- Deux voisinages du point : (a) 8-connexité, (b) 4-connexité

La métrique d_8 (chessboard metric) est définie par :

$$d_8(P, Q) = \max(|i_p - i_q|, |j_p - j_q|)$$

Elle permet d'associer à P, un ensemble de points $v_8(p)$, appelé voisinage et défini par :

$$v_8 = \{Q \in S, d_8(P, Q) \leq 1\}$$

Ce voisinage est représenté sur la figure (2.3.a). Le point P possède 8 voisins dont quatre d'entre eux correspondent à d_4 . On dit que v_8 est un voisinage 8-connexe.

La distance euclidienne entre les pixels $p(x,y)$ et $q(s,t)$ est définie comme suit:

$$D_e(x, y) = \sqrt{(x - s)^2 + (y - t)^2}$$

II.2.14 Intérieur et Extérieur :

Soit S une région, et P un point.

- P est à l'intérieur de S si $\forall Q \in S, \exists$ un chemin connexe entre P et Q tels que tous les points appartiennent à S.
- P est à l'extérieur s'il n'existe pas de chemin connexe dans S entre P et Q.

II.2.15 Négatif

Le négatif d'une image montre les couleurs inverses de l'image initiale (le complément de l'image initiale). Dans l'image à niveaux de gris, on doit soustraire chaque intensité à 255.

Si g est le niveau de gris d'un pixel dans l'image originale, alors le niveau de gris de ce même pixel dans son négatif est 255-g.



Figure2.4- A droite le négatif de l'image originale.

II.2.16 Binarisation d'images

C'est l'opération qui consiste à attribuer deux niveaux de gris (noir et blanc) à une image à plusieurs niveaux de gris, les algorithmes de binarisation sont basés sur les techniques de seuillages comme le montre l'équation suivante :

$I_n(x, y)$ et $I_2(x, y)$ étant les images avant et après binarisation respectivement.

$$I_2(x, y) = \begin{cases} 0 & \text{SI } I_n(x, y) < T \\ 1 & \text{SI } I_n(x, y) \geq T \end{cases}$$

Si $I_n(x, y)$ est inférieur au seuil T alors $I_2(x, y)$ prend la valeur 0 (noire), sinon $I_n(x, y)$ prend la valeur 1 (blanc ou 255).

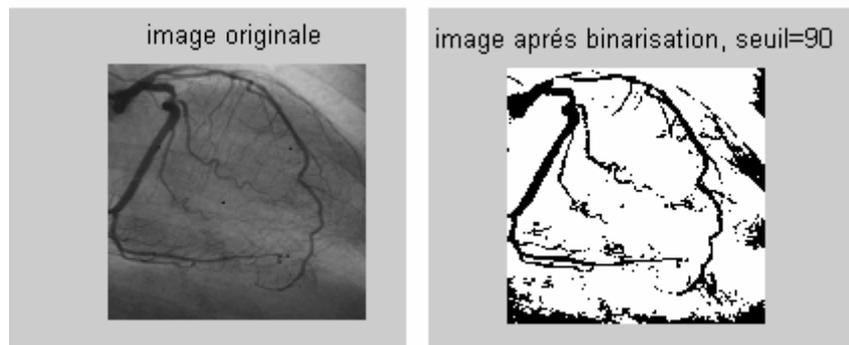


Figure 2.5-Binarisation de l'image, seuil choisi expérimentalement.

II.2.17 Amélioration de contraste

Ce traitement ne s'applique qu'aux images en niveaux de gris. On considère qu'une image est bien contrastée si les luminosités des pixels de l'image sont au mieux réparties sur l'axe [0; 255], en effet deux objets ayant des niveaux de gris très proches, sont difficilement distingués, l'une des méthodes d'amélioration de contraste est la transformation linéaire d'histogramme.

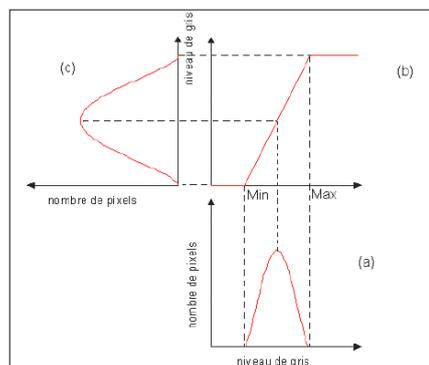


Figure 2.6-Modification d'histogramme

Son principe est de déterminer les bornes maximale et minimale de l’histogramme associées à l’image, et d’appliquer la formule suivante pour chaque pixel de l’image.

$$F'(x, y) = \frac{255}{Max - Min}(F(x, y) - Min)$$

Max : la borne maximale de l’histogramme.

Min : la borne minimale de l’histogramme.

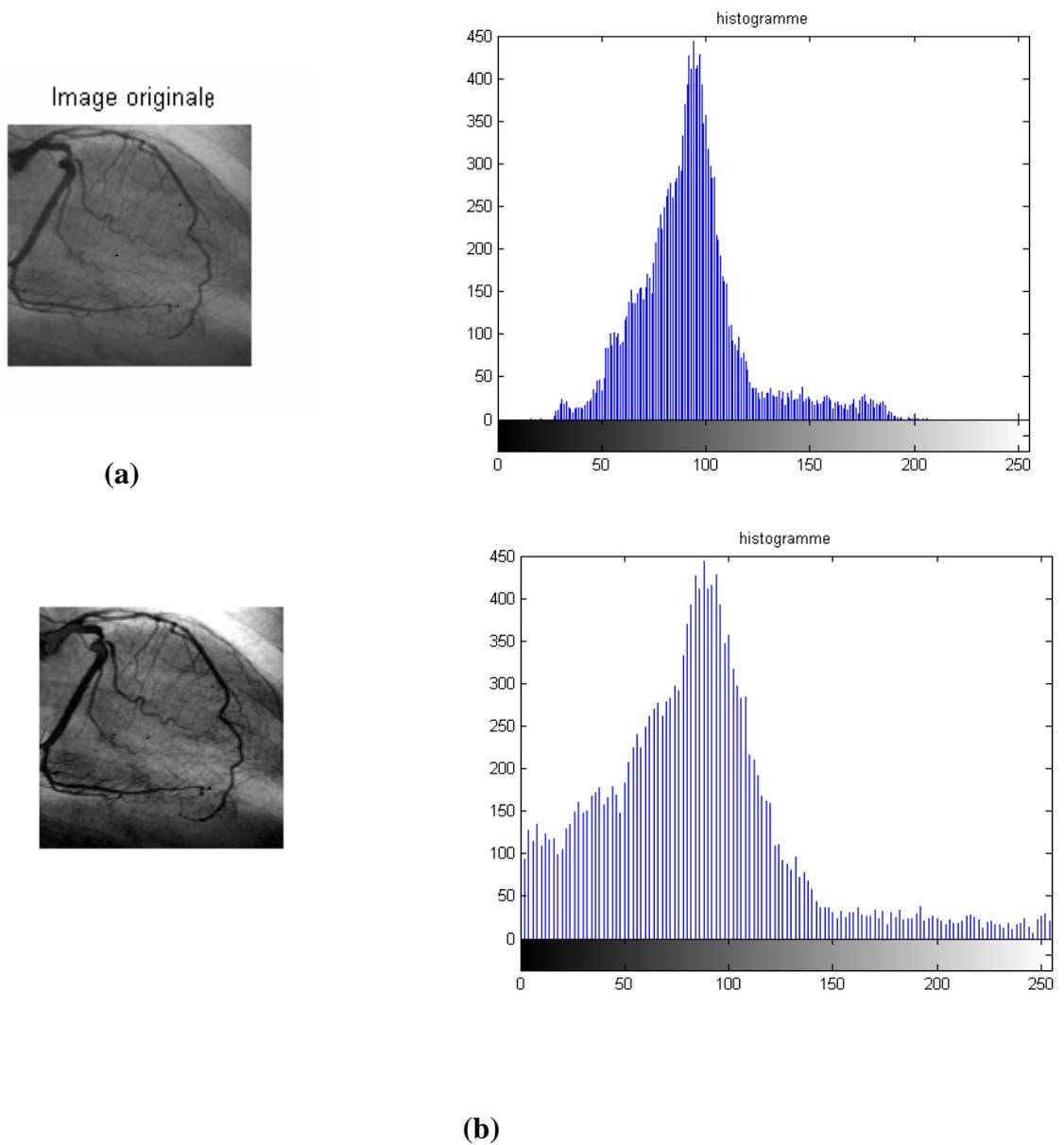


Figure 2.7- (a)Image d’angiographie initiale et (b) image traitée par transformation linéaire.

II.3 Filtrage des images numériques

Nous avons défini le bruit comme un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins. Or une image possède une certaine redondance spatiale : les pixels voisins ont généralement les mêmes caractéristiques. Il est donc nécessaire d'opérer des transformations qui pour chaque pixel tiennent compte de son voisinage. Les méthodes parmi les plus utilisées sont les techniques dites de filtrage de l'image.

Nous distinguons :

- Le filtrage linéaire ;
- Le filtrage non linéaire ;
- Le filtrage Homomorphique ;
- Le filtrage Morphologique.

II.3.1 Filtrage linéaire

Même s'il apparaît rapidement limité dans ses performances, le filtrage linéaire reste très utilisé par sa simplicité d'analyse et de mise en oeuvre. Il constitue en général une référence à laquelle on compare les filtres non linéaires de structure plus complexe. C'est pourquoi nous en rappelons la définition et en illustrons quelques propriétés.

Un filtre linéaire est défini par le fait qu'il satisfait au principe de superposition [4]. Il est décrit généralement par les coefficients de sa réponse impulsionnelle h .

II.3.1.1 Filtre moyenneur

C'est un filtre passe bas, donc il a un effet de lissage sur l'image, ce qui signifie qu'il élimine les signaux de haute fréquence, ce filtrage s'accompagne d'une perte des détails très fines de l'image accompagnée d'une désaccentuation des contours rendant l'image floue ; plus la dimension du filtre augmente, donc N augmente, plus ces effets sont ressentis.

Sa réponse impulsionnelle est donnée par :

$$h(x, y) = \frac{1}{N^2} \quad x, y = 1, \dots, N$$

Pour $N=3$ on aura le masque 3×3 suivant :

$$H_3 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Donc il affecte au pixel central, sa moyenne avec les pixels de la 8-connexité, comme suit.

$$I'(i, j) = \frac{1}{9} \{ I(x-1, y-1) + I(x-1, y) + I(x-1, y+1) + I(x, y-1) + I(x, y) + I(x, y+1) + I(x+1, y-1) + I(x+1, y) + I(x+1, y+1) \}.$$

En général et pour une dimension impaire N du masque on a la formule :

$$I'(x, y) = \frac{1}{N^2} \sum_{i=\frac{1-N}{2}}^{\frac{N-1}{2}} \sum_{j=\frac{N-1}{2}}^{\frac{N-1}{2}} I(x+i, y+j)$$

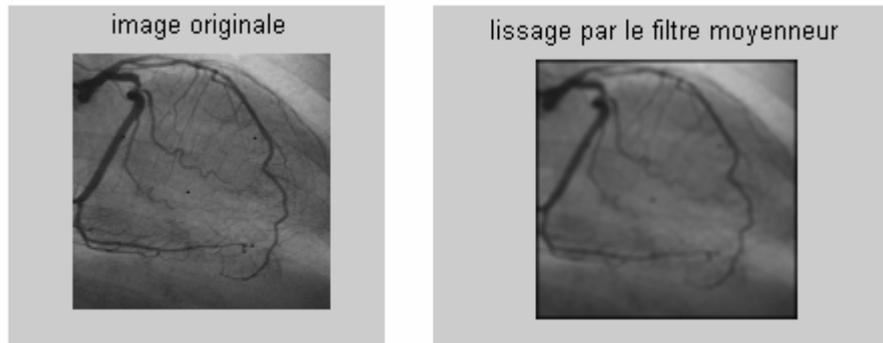


Figure 2.8-Application du filtre moyennneur 3x3.

II.3.1.2. Filtre smooth

Les coefficients du masque pour un filtre 3×3 sont :

$$H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Les coefficients sont calculés en utilisant des pondérations gaussiennes. Des itérations successives permettent d'obtenir le smooth 5×5 (2 itérations) et le smooth 7×7 (3 itérations). Dans ce cas aussi, l'effet du filtre augmente avec la taille de son masque. Les contours et les détails fins sont cependant mieux conservés qu'avec le moyennneur. En effet, en utilisant une pondération gaussienne, le filtre smooth prend mieux en compte les corrélations entre pixels, notamment pour une texture (la fonction de corrélation des niveaux de gris pour une texture est fréquemment modélisée par une gaussienne).

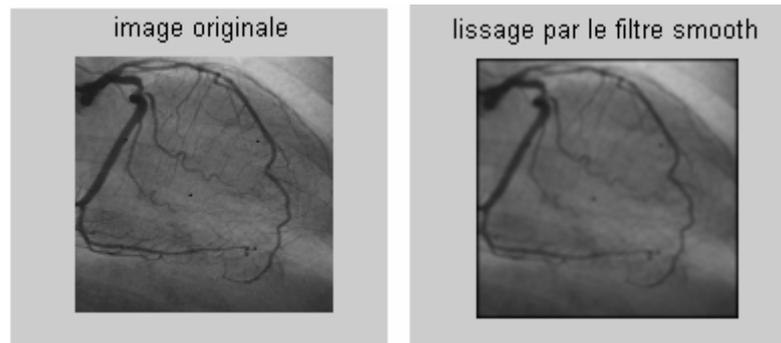


Figure 2.9- application du filtre smooth

Le filtre smooth est un bon exemple des performances qu'on peut obtenir avec un filtre linéaire à réponse impulsionnelle finie. Le gros avantage de ces filtres est leur facilité de conception et d'implémentation, mais ils ne peuvent être utilisés pour des travaux trop fins (la détérioration des contours qu'ils induisent par exemple, empêchera une segmentation fine des images). Ces limitations ont donc conduit à la conception de filtres non-linéaires.

II.3.2 Filtrage linéaire adaptatif

II.3.2.1 Généralités

En tant que processus, l'image présente un caractère non stationnaire marqué. Il est donc naturel d'envisager de traiter les images à l'aide d'opérateurs adaptatifs. Le but est d'obtenir, selon la position dans l'image :

- soit un comportement passe-bas (lissage) ;
- soit un comportement de filtre identité (préservation des frontières) ;
- soit même un comportement passe-haut (rehaussement de contraste).

Bien que la structure de base de ces opérateurs soit celle d'un filtre linéaire, leur fonctionnement dépend d'informations extraites localement leur faisant perdre la propriété de superposition.

L'idée de base est de sélectionner, dans la fenêtre d'analyse, un sous-ensemble de pixels appartenant à la même région que le pixel courant et donc à en exclure ceux appartenant à une région adjacente [4].

II.3.2.2 Moyenne adaptative

Les paramètres du filtre sont les dimensions de la fenêtre d'analyse, $N \times N$ et le seuil de similarité T entre intensités.

Pour une fenêtre d'analyse comportant N^2 pixels, le fonctionnement du filtre est décrit par :

$$I'(x, y) = \frac{\sum_{i=\frac{1-N}{2}}^{\frac{N-1}{2}} \sum_{j=\frac{1-N}{2}}^{\frac{N-1}{2}} C_{i,j} I(x+i, y+j)}{\sum_{N^2} C_{i,j}}$$

$$C_{i,j} = 1 \text{ Si } |I(x, y) - I(x+i, y+j)| < T \text{ Sinon il est nul.}$$

Il s'agit donc de faire la moyenne des intensités des pixels proches, dans l'espace des mesures, du pixel courant. Le seuil doit être choisi suffisamment grand pour tolérer les fluctuations statistiques à l'intérieur d'une région. Il doit être plus faible que l'amplitude des transitions entre régions.

L'effet de réduction de bruit est celui d'un filtre moyenneur quand le seuil est élevé. Le filtre ne lisse pas le bruit impulsionnel sauf si l'on impose un nombre minimal de coefficients non nuls. Il n'y a pas d'effet de rehaussement de contraste.

II.3.3 Filtrage non linéaire

Ces opérateurs ont été développés pour pallier aux insuffisances des filtres linéaires : principalement la mauvaise conservation des contours. Ils ont le défaut d'infliger des déformations irréversibles à l'image [11].

II.3.3.1 Filtrage de Nagao

Le principe de ce filtre est de rechercher, dans la fenêtre d'analyse, une sous fenêtre homogène et de faire la moyenne des intensités des pixels appartenant à cette sous fenêtre. La taille de la fenêtre d'analyse est 5×5 . On y détermine 9 sous fenêtres, appelées aussi **domaines de Nagao** (figure 2.10). Pour chaque domaine D_k , on calcule la moyenne m_k et la variance V_k . La sortie du filtre est la valeur moyenne du domaine ayant la plus faible variance.

Le filtre de Nagao atténue fortement le bruit impulsionnel (d'un facteur 7 à 9) [4]. Il existe des **variantes** à ce filtre, concernant sa taille, le choix du critère d'homogénéité et la non prise en compte du domaine D9.

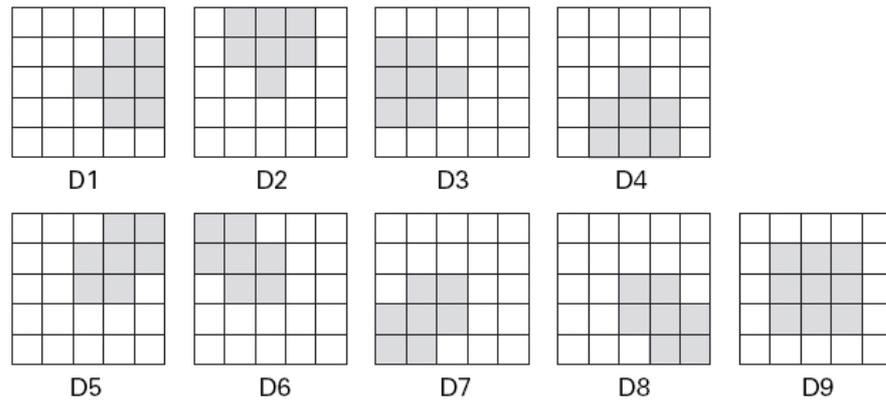


Figure 2.10-Domaines de Nagao.

II.3.4 Filtrage morphologique

Le filtrage morphologique repose sur la morphologie mathématique, fondée sur une description ensembliste des images. Contrairement aux opérateurs précédemment présentés, on privilégie la notion de forme par rapport aux informations sur les amplitudes des variations d'intensité. À la place des opérations d'addition et multiplication, les fonctions de base sont l'« inf » (minimum) et le « sup » (maximum).

Ce type de filtrage est utilisé pour éliminer des pixels isolés dans des images binaires (2 niveaux de gris : noir (0) et blanc (1)) qui sont considérés comme du bruit. Il met en correspondance chaque pixel et son voisin par une fonction logique (ET, OU, XOR). Parmi les opérateurs morphologiques, nous citons

- L'érosion ;
- La dilatation ;
- L'ouverture ;
- La fermeture.

II.3.4.1 L'érosion

C'est l'opération qui consiste à affecter au pixel central le plus petit des éléments du masque. Elle entraîne des effets habituels tels que : la séparation des objets à l'endroit des étranglements, le rétrécissement des objets de grande taille et la disparition des petites composantes. Dans l'exemple ci-dessous, le pixel en question prend la valeur 138 vu qu'elle est la plus petite [9].

178	182	230
148	230	230
138	146	149

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image, on effectue le **OU** logique des huit voisins du pixel considéré.



Figure 2.11-Image érodée après inversement des niveaux de gris

II.3.4.2 La Dilatation

C'est l'opération qui consiste à affecter au pixel central le plus grand des éléments du masque. Elle permet d'éliminer les pixels blancs isolés. On effectue le **ET** logique des huit voisins du pixel considéré. Elle élimine les tâches blanches dans des zones noires mais ajoute des pixels noirs au contour des objets présents dans l'image [9].



Figure2.12- Image dilatée après inversement des niveaux de gris

II.3.4.3 L'Ouverture

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet de retrouver la taille normale des objets de l'image [9].



Figure 2.13- Ouverture de l'image après inversement des niveaux de gris.

II.3.4.4 La Fermeture

La fermeture est une opération morphologique qui consiste à faire subir à l'image une opération de dilatation suivie d'une érosion. Elle permet de retrouver la taille normale des objets de l'image.



Figure 2.14- Fermeture de l'image après inversement des niveaux de gris.

II.3.5 Filtrage homomorphique

Ce filtrage tire son nom de la propriété mathématique d'homomorphisme, c'est-à-dire de préservation de la structure de l'espace des données [9].

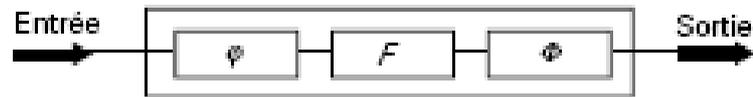


Figure 2.15-Filtrage homomorphique.

En pratique, on cherche, par une transformation appropriée, à rendre additif l'effet du bruit d'image puis à réduire l'influence de ce bruit par un filtrage linéaire efficace.

Le schéma bloc d'un filtre homomorphique comporte trois étages (figure 2.15). Les opérateurs φ et ϕ sont inverses l'un de l'autre :

$$\varphi * \phi = I$$

F est un filtre linéaire.

II.4 CONCLUSION

Dans ce chapitre nous avons présenté quelques définitions liées au traitement d'images, ainsi qu'une phase de son processus à savoir le prétraitement fondé sur les algorithmes de filtrage classiques et adaptatifs. Ces derniers, basés sur des attributs stochastiques et des transformations ponctuelles, présentent davantage d'améliorations dans la réduction du bruit introduit lors de l'acquisition.

CHAPITRE III

SEGMENTATION

D'IMAGES

Chapitre III

SEGMENTATION D'IMAGES

III.1 INTRODUCTION

La segmentation est l'opération qui consiste à regrouper les points images partageant une certaine propriété pour former une région (approche région), ou à identifier les points connexes qui présentent une forte transition d'intensité (détection de contour), cette dernière opération est parfois traitée à part, dans littérature, indépendamment de la segmentation.

Le but de la segmentation est d'extraire les informations pertinentes en regard de l'application concernée. Ce processus se situe à la frontière des traitements bas-niveaux (détection de contour) et des traitements haut-niveaux (approche région).

IL n'existe pas d'algorithme qui donne satisfaction dans le cas général, le choix de la technique à utiliser est lié à la nature de l'image et aux opérations situées en aval de la segmentation.

III.2 APPROCHE FRONTIERE OU DETECTION DE CONTOURS

L'extraction de contours joue un rôle primordial dans tout système de vision par ordinateur. Beaucoup d'efforts ont été faits pour extraire les contours d'une image et plusieurs techniques ont été proposées à ce jour. Les contours correspondent généralement à des

changements brusques de propriétés physiques ou géométriques de l'image perçue et forment ainsi des attributs très importants pour l'analyse.

III.2.1 Opérateur dérivatif du premier ordre

La première approche possible pour détecter les variations locales de la fonction $f(x,y)$ représentant l'intensité est d'utiliser une transformation du type gradient.

Le gradient d'une image est obtenu en appliquant le vecteur suivant [14]:

$$\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t \quad (3.1)$$

Ce vecteur peut être représenté par son module m (relation 3.2) et sa phase ϕ (relation 3.3) qui donne la direction :

$$m = \sqrt{\left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2} \quad (3.2)$$

$$\phi = \arctan \frac{\frac{\partial I(x, y)}{\partial x}}{\frac{\partial I(x, y)}{\partial y}} \quad (3.3)$$

D'après la relation (3.4) il s'agit, dans le cas discret, de calculer les différences des luminances suivant x et y [6].

$$\frac{\partial I}{\partial x} \approx \frac{\Delta I}{\Delta j} = I(i, j+1) - I(i, j) \quad (3.4.a)$$

$$\frac{\partial I}{\partial y} \approx \frac{\Delta I}{\Delta i} = I(i+1, j) - I(i, j) \quad (3.4.b)$$

Pour des commodités de calculs, on prend :

$$m = \left| \frac{\Delta I}{\Delta i} \right| + \left| \frac{\Delta I}{\Delta j} \right| \quad (3.5.a)$$

Ou encore

$$m = \max \left(\left| \frac{\Delta I}{\Delta x} \right|, \left| \frac{\Delta I}{\Delta y} \right| \right) \quad (3.5.b)$$

Rappelons la convention d'orientation des axes :



Notons que ces opérateurs sont particulièrement sensibles au bruit [8].

III.2.1.1 Opérateurs de Roberts

C'est un détecteur de contours qui calcule les dérivées suivant les transitions diagonales, les masques de Roberts sont définis par les matrices suivantes :

$$h1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Le calcul s'effectue de la même façon que celui de l'opérateur gradient, mais qui doit être précédé par un filtrage [8].



Figure 3.1-Opérateur de Robert (à droite) appliqué à l'image originale.

III.2.1.2 Opérateur de Prewitt et de Sobel

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

$$I_j[i, j] = h_j * I[i, j] \quad \text{et} \quad I_i[i, j] = h_i * I[i, j] \quad (3.6)$$

Avec

$$h_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \quad \text{et} \quad h_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix}$$

Les matrices h_i et h_j , appelées aussi masques, sont les noyaux de convolution de filtres à réponse impulsionnelle finie.

Les masques de Prewitt sont définis pour $c=1$ et les masques de Sobel pour $c=2$. Rappelons que la relation entre convolution et corrélation peut être traduite par l'expression :

$$\begin{aligned} h * I[i, j] &= \sum_{m=-M}^M \sum_{n=-N}^N h[m, n].I[i-m, j-n] \\ &= \sum_{m=M}^{-M} \sum_{n=N}^{-N} h[-m, -n].I[i+m, j+n] \end{aligned}$$

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 » $h_j[-j]$ et $h_i[-i]$ [6].

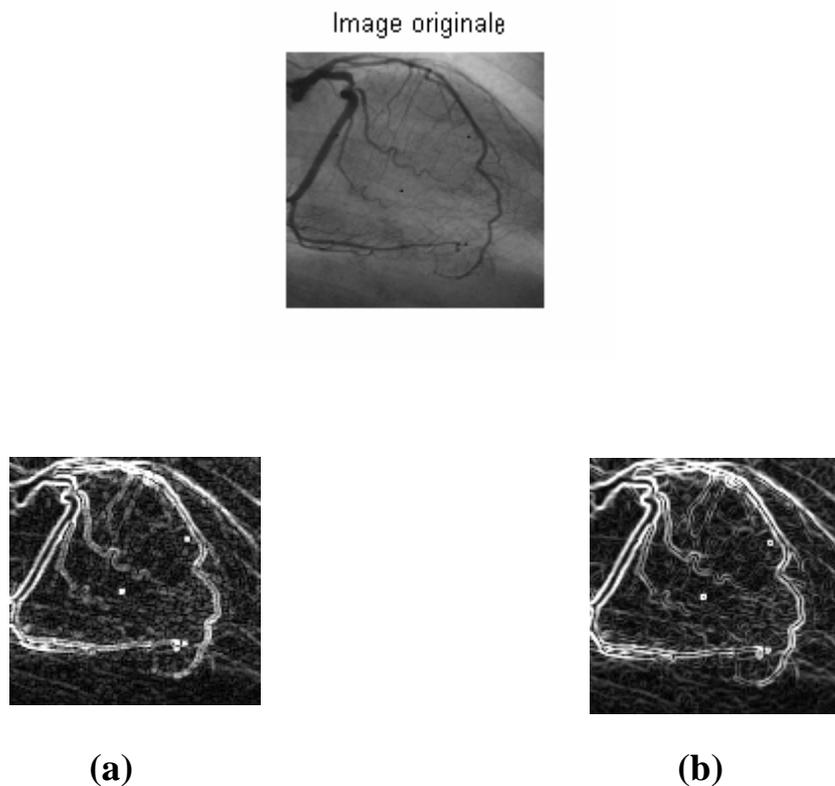


Figure 3.2-Application des opérateurs de Prewitt (a) et de Sobel (b) à l'image originale.

III.2.1.3. Opérateur MDIF et NAGDIF

Ces deux opérateurs sont la combinaison d'un lissage et d'une dérivation. Le premier est linéaire et le second est non linéaire.

Opérateur MDIF

Il s'agit de la combinaison d'un filtre moyenneur de noyau m et d'un dérivateur utilisant les masques directionnels de Prewitt h_1, h_2 [6].

$$m = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h_j = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad h_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.7)$$

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

$$m_j = m * h_j = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix} \quad m_i = m * h_i = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (3.8)$$



Figure 3.3- Application de l'opérateur MDIF sur l'image originale

Opérateur NAGDIF

L'opérateur NAGDIF est la combinaison d'un lissage non linéaire de Nagao suivi d'une opération de dérivation effectuée par l'opérateur suivant :

$$\text{CONTOUR SI } \max\{B[i, j] - B[i - \varepsilon_1, j - \varepsilon_2]\} > \text{seuil}$$

ε_1 et ε_2 sont des valeurs entières de l'intervalle $[-1,1]$ et B est l'image lissée.

Les seuils sont réglés empiriquement, L'opérateur NAGDIF donne des contours fins mais moins réguliers que l'opérateur MDIF.

Les deux opérateurs sont particulièrement intéressants dans des images peu bruitées [6].

III.2.1.4 Opérateurs gradients directionnels : Kirsh, 3 et 5 niveaux

L'opérateur de Kirsh est un opérateur à huit masques correspondant chacun à une direction préférentielle et obtenu par rotation de $\frac{\pi}{4}$ de l'opérateur de base h_0 :

$$h_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}, \quad h_1 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix}, \dots \quad (3.9)$$

Le gradient retenu sera celui correspondant à la valeur maximum donnée par :

$$\max_i = \lfloor h_i * I \rfloor_{1,8}.$$

L'orientation retenue pour le contour sera celle correspondante au masque ayant permis d'obtenir le gradient maximum [8].

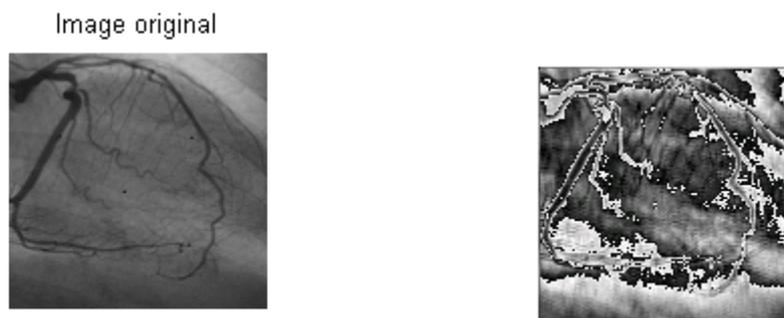


Figure 3.4- application de l'opérateur de Kirsh.

III.2.2 Opérateurs dérivatifs du second ordre

Le Laplacien d'un signal continu est donné par la relation suivante [15]:

$$\Delta I(x, y) = \frac{\partial^2 I(x, y)}{\partial^2 x} + \frac{\partial^2 I(x, y)}{\partial^2 y} \quad (3.10)$$

En discret, l'approximation la plus simple du Laplacien est donnée par l'application de l'un des masques suivants :

$$h_4 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad h_8 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad h = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \dots$$

Les points de contours correspondent alors aux passages par zéro de l'image obtenue par convolution avec l'opérateur Laplacien. Une opération de seuillage sur la norme du gradient est généralement nécessaire afin d'éliminer les contours correspondant au bruit [8].

III.2.3 Gradient morphologique

On peut obtenir un contour en calculant la différence entre la dilatation d'une image donnée et son érosion [4], relation (2.7) :

$$C(i, j) = \text{Dilatation}(A(i, j)) - \text{Erosion}(A(i, j))$$

Le calcul de la norme du gradient est très sensible au bruit. Cette approche ne peut s'appliquer que sur une image lissée



Figure 3.5- application du gradient morphologique après un lissage par le filtre moyennur.

III.3 Approche région

L'approche duale de la détection des contours pour la décomposition d'une image en ses formes élémentaires est l'approche par régions. Elle repose sur la recherche de zones possédant des attributs communs, soit de luminosité, soit, plus rarement, de textures.

La segmentation en régions homogènes vise à segmenter l'image en se basant sur les propriétés intrinsèques de la région, le choix de ces propriétés détermine ce que l'on appelle le critère de la segmentation.

III.3.1 Segmentation par seuillage

Elle consiste à comparer l'attribut de chaque point de l'image à un ou plusieurs seuils en vue d'attribuer le point considéré de l'image à une certaine classe, cette définition est formalisée de la manière suivante :

$$P_i, i = 1, \dots, I \in C_l, l = 1, \dots, L \quad \text{Si} \quad T_l < A(P_i) < T_{l+1}$$

Dans cette relation, P_i représente les points numéro i de l'image originale et C_l représente la classe numéro l . La classification de chaque point de l'image s'effectue en comparant son attribue $A(P_i)$ à un ensemble de L seuil T_l , avec $l = 1, \dots, L$. Par exemple dans le cas de la segmentation d'une image en deux classes objet et fond, la relation se simplifie et devient [5] :

$$P_i = 1, \dots, I \in \text{« Objet »} \quad \text{Si} \quad g(P_i) < T$$

III.3.2 Traitement à multirésolution

Parmi les nombreuses méthodes de segmentation existantes, il y a l'approche multirésolution. Elle permet en effet de mettre en oeuvre le principe de focus d'attention : « pour chaque entité de l'image, il existe une résolution donnée dans laquelle la recherche et l'extraction sont facilitées », cette échelle est reliée aux éléments structurels constituant l'image.

III.3.2.1 Définition de la pyramide

Commençons par fixer quelques notations [16]

Une image I sera un ensemble de pixels $p(x, y)$ tel que $I = [0, 2^n - 1] \times [0, 2^n - 1]$.

A chaque pixel sera associé un niveau de gris $f(p) = f(x, y)$ tel que $\forall p \in I; 0 < f(p) < M$
 M est le nombre de niveaux de gris dans l'image.

Le voisinage de taille N d'un pixel $p(x, y)$ sera :

$$V_{N(p)} = \{q(x, y) \in I; |x - x_p| < N \text{ et } |y - y_p| < N\}$$

Une pyramide P est une suite d'images I_h , chacune correspondant à une version de l'image initiale I_0 pour une résolution réduite. Chaque niveau h est construit par une convolution du niveau inférieur avec un noyau de taille k .

Généralement, la taille est divisée par 2 à chaque niveau si bien que

$$I_h = [0, 2^{n-h} - 1] \times [0, 2^{n-h} - 1].$$

I_0 et T sont respectivement la base et la hauteur de la pyramide.

Si $f_h(x, y)$ désigne le niveau de gris de l'élément (x, y) de l'image I_h constituant la pyramide. f_0 Représente les niveaux de gris de l'image initiale.

Chaque élément du niveau h de la pyramide est construit par l'intervention de k^2 éléments du niveau inférieur, ses fils. En fait,

$$f_h(x, y) = \sum_{i=1}^{i=k} \sum_{j=1}^{j=k} f_{h-1}(2x + i - z, 2y + j - z) w(i, j)$$

Où z est une constante valant la partie entière $\frac{k+1}{2}$. Chaque élément du niveau h engendre des pères au niveau supérieur, en intervenant dans leur construction.

Cette construction est ainsi définie par son masque de pondération $W = [w(i, j)]$, $i \in [1, k]$, $j \in [1, k]$. C'est généralement un filtre passe-bas, adoucissant considérablement les contours [16].

III.3.2.2 Pyramide gaussienne

Un exemple simple de représentation pyramidale est la pyramide gaussienne. Cette pyramide est construite par répétitions multiples de moyennage local, au fur et à mesure que l'indice du niveau augmente, la résolution du signal diminue, contenant de moins en moins de détails vers le haut de la pyramide [5].

Les différents niveaux correspondent ainsi à un filtrage passe-bas effectué récursivement, suivi d'un sous échantillonnage pour chaque niveau. L'ensemble des filtres est utilisé pour obtenir les différents niveaux de résolution de la pyramide et ils se démarquent les uns des

autres seulement dans l'échelle et non pas dans la forme. La réponse impulsionnelle de ces filtres s'approche de la forme d'une courbe de gauss, d'où le nom de la pyramide.

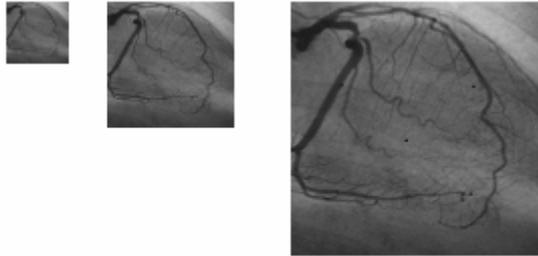


Figure 3.6- Diminution de la résolution d'un facteur de 2 de droite à gauche (128x128- 64x64- 32x 32).

Remarque :

En général, les structures spatiales inhérentes et les tailles des régions ne sont pas connues a priori. En plus, la non stationnarité générale des images fait qu'une échelle de résolution idéale et une taille de fenêtre idéale pour une certaine région ne le sont pas plus pour une autre région dans une même image.

III.4 CONCLUSION

La segmentation est une phase incontournable dans l'analyse d'image. Elle permet d'extraire les informations nécessaires à la localisation et la délimitation des structures d'intérêt dans l'image. Nous avons exposé dans ce chapitre un ensemble de méthodes de segmentation par approche frontière, et par approche région et on a terminé par une introduction au traitement multirésolution.

CHAPITRE IV

SQUELETTISATION

Chapitre IV

SQUELETTISATION

IV.1 INTRODUCTION

La **squelettisation** est une classe d'algorithmes utilisée en analyse de formes. Elle consiste à réduire une forme en un ensemble de courbes, appelées **squelettes**, centrées dans la forme d'origine. La squelettisation est un outil d'analyse de forme qui conserve les propriétés topologiques de la forme d'origine ainsi que les propriétés géométriques, selon la méthode employée. C'est une étape primordiale dans notre travail, car elle nous permettra par la suite d'appliquer l'algorithme de l'étiquetage.

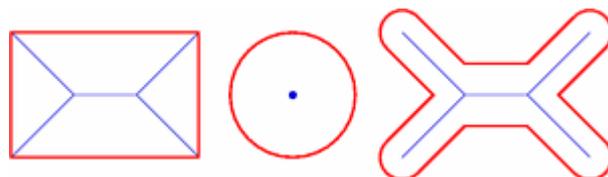


Figure IV.1- Exemples de squelettes pour des formes simples.

IV.2 DEFINITION

La squelettisation est une méthode qui a été développée à l'origine dans les années soixante par Harry Blum, en vue de créer un nouveau descripteur de formes. Cette méthode a gagné l'intérêt de nombreux chercheurs. Actuellement, la squelettisation est une méthode très connue en analyse d'image. Il existe un grand nombre d'algorithmes proposant de transformer une forme en squelette.

En termes simples, la squelettisation consiste à *amaigrir* une forme jusqu'à obtenir un ensemble de courbes centrées. L'ensemble obtenu est alors appelé **squelette** ou **axe médian**.

La définition ci-dessous a été énoncée par Harry Blum et est connue sous le nom *d'analogie du feu de prairie*. Elle offre une vision intuitive de la squelettisation.

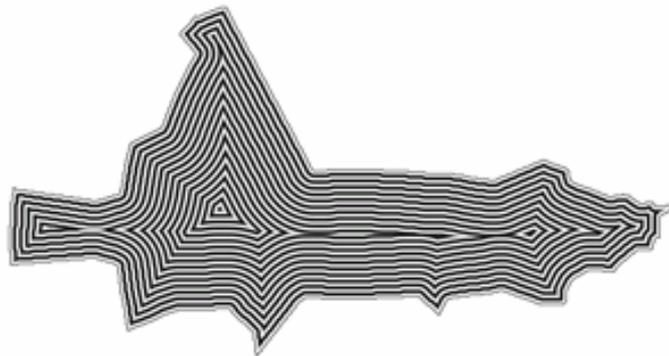


Figure IV. 2- Evolution du front enflammé dans une forme.

Soit une prairie couverte de manière homogène par de l'herbe sèche et Ω un ensemble de points de cette prairie. Au départ, tous les points du contour de Ω sont enflammés simultanément. Le feu se propage de manière homogène et s'étend à travers la prairie à une vitesse constante. le *squelette* de l'ensemble de points Ω (noté $\mathbf{MA}(\Omega)$) est défini comme le lieu des points où les fronts enflammés se sont rencontrés.

IV.3 LES ALGORITHMES DE SQUELETTISATION

Ils sont très variés et leur application dépend souvent du type d'image à traiter.

On peut les classer en deux types d'algorithmes :

- à critère topologique
- à critère métrique

IV.3.1 LES ALGORITHMES A CRITERE TOPOLOGIQUE

Appelés aussi les algorithmes de « pelage » (thinning algorithms) [22]. Ce sont des algorithmes itératifs supprimant à chaque étape, le long de la frontière de la forme, les points dits inessentiels (c'est-à-dire n'appartenant pas au squelette). Ils utilisent souvent des fonctions booléennes opérant sur des voisinages de points, déterminant à chaque passage la validité des points frontières.

```
début  
  répéter  
    inessentiel ← faux  
    pour tout point dans l'image faire  
      si point inessentiel(point) alors  
        inessentiel ← vrai  
        image(point) ← 0  
      fsi  
    fpour  
      jusqu'à inessentiel = faux  
fin
```

Il existe plusieurs algorithmes à critère topologique. Nous en avons sélectionné 3 des plus connus :

- L'algorithme de Zhang et Suen (qu'on a choisi pour détailler)
- L'algorithme de Rosenfield
- L'algorithme de Serra

IV.3.1.1 L'algorithme de Zhang et Suen

Dans cette section nous présentons un algorithme développé par Zhang et Suen pour amincir des images binaires. Les pixels des objets ont une valeur égale à 1 (noir), et les pixels du fond 0 (blanc). Et nous considérons à chaque fois un voisinage de huit pixels, comme illustré dans la figure :

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure IV.3 : Le voisinage du pixel P1.

Cet algorithme comporte deux étapes, dans chacune d'elles le pixel P1 et ses huit voisins doivent vérifier un ensemble de conditions.

La première étape force à zéro le pixel P1 du contour, qui satisfait toutes les conditions suivantes :

- 1) $2 \leq N(p1) \leq 6$
- 2) $S(p1) = 1$
- 3) $P2 * p4 * p6 = 0$
- 4) $P4 * p6 * p8 = 0$

Tel que : $N(p1) = p2 + p3 + p4 + \dots + p9$ est le nombre de voisins noirs du pixel p1.

Et $S(p1)$ est le nombre de transitions 0-1 dans l'ordre $p2, p3, \dots, p9$.

La deuxième étape garde les deux premières conditions, mais les conditions 3 et 4 deviennent :

- 3') $p2 * p4 * p8 = 0$.
- 4') $p2 * p6 * p8 = 0$.

Les pixels qui vérifient les conditions des étapes 1 et 2 sont forcés à zéro (blanc), et ceux qui ne les vérifient pas gardent leurs mêmes valeurs.

On note que les pixels vérifiant ces conditions doivent être supprimés simultanément. L'algorithme de Zhang et Suen est donné comme suit :

DEBUT

Pour chaque pixel P1 de l'image obtenue à l'étape de binarisation **faire**

DEBUT

$N(p1) = p2+p3+p4+p5+p6+p7+p8+p9$; { nombre de voisins noirs de p1 }

$S(p1) =$ nombre de transitions 0-1 dans l'ordre $p2, \dots, p9, P2$;

{1^{er} étape :}

Si ($(2 \leq N(p1) \leq 6)$ et $(S(p1)=1)$ et $(p2*p4*p6=0)$ et $(p4*p6*p8=0)$) **alors**

Enregistrer tous les points qui satisfont les conditions ;

{2^{ème} étape :}

Si ($(2 \leq N(p1) \leq 6)$ et $(S(p1)=1)$ et $(p2*p4*p8=0)$ et $(p2*p6*p8=0)$) **alors**

Enregistrer encore tous les points qui satisfont les conditions. ;

Si tous les points qui satisfont les conditions ont été détectés **alors**

Forcer à zéro tous les points enregistrés;

FIN ;

FIN.

L'étape 1 est appliquée sur chaque pixel de la frontière de l'objet dans l'image binaire. Si une condition ou plusieurs de cette étape ne sont pas satisfaites, la valeur du pixel en question reste inchangée. Si toutes les conditions sont satisfaites, le point est signé pour une éventuelle suppression. Il est important de noter que le point n'est pas supprimé simultanément mais jusqu'à ce que tous les points contours soient balayés. Ceci efface le risque de changer la structure de l'image lors d'une itération dans le traitement. Après que l'étape 1 soit appliquée à tous les points contours de l'image, les points signés sont alors supprimés (changés en 0).

L'étape 2 est ensuite appliquée au résultat obtenu de la même manière que l'étape 1.

La condition (1) n'est pas satisfaite lorsque le point contour p_1 possède un seul ou sept pixel voisin, parmi le 8-voisinage, de valeur 1. Possédant un seul voisin de valeur 1 implique que le point p_1 est un point terminal dans le squelette et ne sera pas donc supprimé.

Si le point p_1 possède 7 voisins de valeurs 1 et sera aussi supprimé, on obtiendra une érosion dans la région. La condition (2) ne sera pas satisfaite si elle est appliquée aux pixels d'un squelette mince. Donc, cette condition préserve la connexion des segments du squelette durant

l'opération de suppression. Les conditions (3) et (4) sont satisfaites simultanément par les valeurs minimales suivantes : $p_4 = 0$, ou $p_6 = 0$, ou ($p_2 = 0$ et $p_8 = 0$). Avec l'arrangement des voisins du pixel donné dans la figure (5), un point qui satisfait ces conditions, ainsi que les conditions (1) et (2), est un point frontière est ou sud, ou un coin nord-ouest dans la frontière.

Dans les autres cas, le point p_1 ne fait pas partie du squelette et est donc supprimé. De façon similaire, les conditions (3') et (4') sont satisfaites par les valeurs minimales suivantes : $p_2 = 0$, ou $p_8 = 0$, ou ($p_4 = 0$ et $p_6 = 0$). Ceci correspond aux points nord ou ouest de la frontière, ou un coin sud-est.

IV.3.1.2 L'algorithme de Serra

Pour déterminer si un point est inessentiel, Serra [14] utilise les masques suivants :

$$\begin{array}{ccc} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{array} \quad \text{et} \quad \begin{array}{ccc} x & 0 & 0 \\ 1 & 1 & 0 \\ x & 1 & x \end{array}$$

L'ensemble des voisinages 3*3 d'un point inessentiel est obtenu à partir de ces configurations par rotation de 90°.

L'algorithme général est de type séquentiel. Il consiste à balayer l'image ligne par ligne et à supprimer les points inessentiels au fur et à mesure qu'ils sont rencontrés. Il ne peut donc pas tenir compte de l'environnement dans lequel il opère et a tendance à enlever trop de points d'un seul coup.

Un autre procédé, de type parallèle, consiste à décomposer la suppression en sous cycles permettant d'approcher le squelette par raffinement successif. Dans l'exemple suivant, tous les points sont des points frontières et peuvent être considérés, chacun à part, inessentiels :

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

Si l'on supprime d'abord les deux points de dessous (égaux à 1), on s'aperçoit que les points restants sont essentiels et ne doivent pas être supprimés. L'idée de l'algorithme parallèle est donc d'imposer progressivement des contraintes sur les points de manière à dégager petit à petit le vrai squelette.

IV.3.1.3 L'algorithme de Rosenfeld

L'algorithme de Rosenfeld détermine le degré d'érosion de chaque point de l'objet. Cet algorithme permet la squelettisation en 4 ou 8-connexité.

- ✓ Soit X l'objet obtenu après m érosions sur l'objet initial.
- ✓ Soit Y l'objet obtenu après m+1 érosions sur l'objet initial.
- ✓ Soit Z l'objet obtenu par différence entre X et Y.

Le squelette de l'image source selon la méthode proposée par Rosenfeld est défini comme la différence entre l'image source et la réunion des m objets Z .

Autrement dit, un point appartient au squelette si c'est un maximum local pour les degrés d'érosion, c'est à dire qu'il ne peut être reconstruit par aucun de ses voisins. La construction du squelette suivant la méthode de Rosenfeld est réversible.

Il est clair que les degrés d'érosion calculés dépendent de la connexité du masque. On en déduit qu'il existe nécessairement une différence entre les squelettes obtenus en 4 et 8-connexité.

IV.3.2 LES ALGORITHMES A CRITERES METRIQUES

A l'inverse des algorithmes précédents, on s'intéresse plus ici aux points du squelette, c'est-à-dire aux points essentiels [23]. Ces points sont normalement situés aux centres de la forme, c'est-à-dire à égales distances des frontières. Cette notion de distance est la clé de toutes ces méthodes métriques.

On définit la distance d'un point x à un ensemble de points A par :

$$d(x,A) = \min_{y \in A} d(x,y).$$

Soit B la frontière de la forme, le squelette S est alors défini par :

$$S = \{x / \exists y, z \in B, y \neq z : d(x,B)=d(x,y)=d(x,z)\}.$$

Cette définition de la distance ne doit pas être appliquée de manière stricte car la distance entre les points image n'est pas très précise à cause de l'échantillonnage. Plusieurs méthodes dans la littérature sont fondées sur cette notion de distance. Nous détaillons à titre d'exemple l'algorithme de Marthon.

IV.3.2.1 L'algorithme de Marthon

C'est un algorithme local qui sélectionne les points qui sont des maxima pour leurs distances à la frontière du caractère [24]. Cet algorithme ne considère que les voisinages de type 3×3 . Pour chaque point, il calcule la somme des vecteurs joignant ce point à ses voisins. Les coordonnées des points voisins prennent leurs valeurs dans $[0,+1,-1]$ suivant leurs positions par rapport au point central, considéré comme l'origine. La longueur est calculée en sommant les valeurs absolues des composantes X et Y du vecteur somme. Elle est comprise entre 0 et 4.

Considérons le point p ayant 4 voisins V_1, V_2, V_3, V_4 disposés comme le montre le schéma ci-après :

$$\begin{array}{ccc} V_3 & \bullet & V_2 \\ \bullet & p & \bullet \\ \bullet & \bullet & V_4 \end{array}$$

Dans ce cas, le vecteur somme a pour composantes :

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \end{bmatrix} + \begin{bmatrix} +1 \\ +1 \end{bmatrix} + \begin{bmatrix} +1 \\ 0 \end{bmatrix} + \begin{bmatrix} +1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ +1 \end{bmatrix} \Rightarrow |X| + |Y| = 3.$$

Intuitivement, nous pouvons dire que les points squelette sont associés à une longueur $|X| + |Y|$ petite.

L'algorithme de squelettisation élimine les points dont la valeur est égale à 4 et retient ceux dont la valeur est inférieure ou égale à 2. Les points dont la valeur est égale à 3 nécessitent d'abord une étude du voisinage. Par exemple, pour la configuration suivante :

•		•
	p	•
		•

On a $\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 2 \\ +1 \end{bmatrix}$, d'où $|X| + |Y| = 3$, et p doit être gardé pour éviter une disconnexion de la forme. Par contre dans le

		•
	p	•
		•

cas ci-dessous :

On a $\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$, d'où $|X| + |Y| = 3$, et p doit être éliminé.

Pour l'ensemble des points, la décision peut être prise en incluant une information concernant le nombre de points voisins (nvois). Il y a cependant deux exceptions à cette règle correspondant aux cas de la ligne double ou la colonne double.

⇨ Description de l'algorithme :

L'algorithme est itératif. Chaque itération comprend deux passages par la forme (ou deux balayages de l'image). Le premier passage calcule les caractéristiques locale de chaque point ($|X|, |Y|$ et nvois). Le deuxième élimine ou non les points en fonction de leurs caractéristiques. L'algorithme s'arrête lorsqu'il n'est plus possible d'éliminer des points.

Premier balayage :

```

Pour chaque point p à traiter faire
  Calculer  $|X|, |Y|$  et nvois
   $|X| + |Y| \leftarrow 3$  alors  $|p| \neq$       Si  $|X| + |Y|$ 
    Sinon
      Si nvois=2 ou nvois=6 ou (nvois=3 et  $\max(|X|, |Y|)=3$ )
        ou (nvois=5 et  $\max(|X|, |Y|)=3$ ) alors
          si configuration ligne double ou colonne double
            alors  $|p| \leftarrow 3$ 
            sinon  $|p| \leftarrow 5$ 
          fsi
        sinon  $|p| \leftarrow 3$ 
      fsi
    fsi
  fpour
    
```

Deuxième balayage : élimine tous les points p tel que $|p| \geq 4$.

Le squelette obtenu contient des ébarbures et a parfois une épaisseur supérieure à 1. Un dernier passage par l'image élimine ces défauts en supprimant les points ayant une des configurations suivantes :

$$|X|=1, |Y|=1 \text{ et } n_{\text{vois}}=2,$$

$$|X|=0, |Y|=2 \text{ et } n_{\text{vois}}=3,$$

$$|X|=2, |Y|=0 \text{ et } n_{\text{vois}}=3,$$

IV.4 BINARISATION

Avant de pouvoir appliquer les algorithmes précédents, l'image doit être binarisée. La binarisation consiste en l'obtention d'une image en seulement deux niveaux de gris (0 et 255).

En choisissant un seuil de binarisation S , on peut appliquer l'algorithme suivant :

Pour chaque pixel :

$$I(L(i,j)) = \begin{cases} \text{Noir} & \text{si } L(i,j) > S \\ \text{Blanc} & \text{si } L(i,j) \leq S \end{cases}$$

$L(i,j)$ est le niveau de gris du pixel (i,j)

L'application de cette procédure n'a pas donné de bons résultats, et cela en testant différents seuils de binarisation.

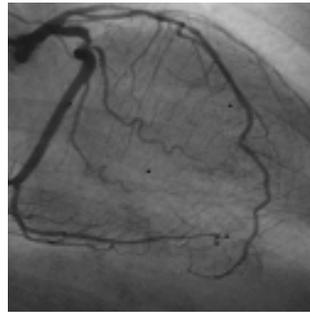


Image Originale



Résultat pour S=75



Résultat pour S=80



Résultat pour S=90

Figure IV.3- Résultats de la binarisation pour différents seuils.

Pour obtenir une image binaire de meilleur qualité, on a appliqué la méthode suivante [14] basée sur l'exploitation des résultats d'un détecteur de contour (Sobel)

- ✓ Une détection de contours par application du masque de Sobel.
- ✓ Un amincissement de l'image résultat.
- ✓ Pour chaque point contour détecté, on considère une fenêtre, de taille dépendante de la dimension de l'artère, sur l'image originale, centré sur ce point.
- ✓ Sur chaque fenêtre, on applique un seuillage local.
- ✓ L'ensemble de ces fenêtres, constitue une binarisation de l'image originale.
- ✓ Pour améliorer le résultat, on peut appliquer un filtrage morphologique sur l'image obtenue.

Algorithme :

Pour chaque point p de l'image **faire**

Si p est un point contour **faire**

Créer une fenêtre centrée sur le point p ,

Seuillage local de la fenêtre.

Fsi

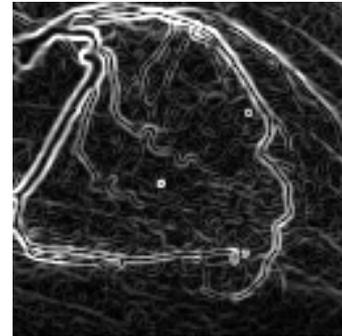
fpour

Pour minimiser les chances de rencontrer un bruit que l'on peut assimiler à un point contour, le balayage de l'image originale se fait avec un pas différent de 1, défini selon la dimension de l'artère.

Résultats de la méthode proposée



Image Originale



Detection de contours par Sobel



Image Binarisée



Image filtrée (médian)

Figure IV.4- Amélioration de la binarisation.

IV.4 APPLICATION

Afin d'obtenir un squelette adéquat pour la suite de notre travail, nous avons choisi d'appliquer deux méthodes, une à critère métrique et l'autre à critère morphologique, et d'opter pour celle qui donnera le squelette le mieux adapté à notre algorithme d'étiquetage.



Image originale



Image binarisée

**Zhang et Suen****Marthon****Figure IV.5-** Squelettes obtenu par différents algorithmes

On remarque que les deux algorithmes suivent bien la direction de l'objet, mais l'algorithme de Zhang et Suen présente une multitude de points isolés, ainsi que des discontinuités qui seront un grand obstacle dans l'étiquetage par la suite. Par contre l'algorithme de Marthon donne un squelette sans trop de discontinuités, c'est cet algorithme qu'on a choisi pour la suite de notre travail.

IV.5 CONCLUSION

Dans ce chapitre nous avons défini une étape primordiale dans notre travail : la squelettisation. Nous avons détaillé quelques algorithmes et appliqué deux d'entre eux, un à critère topologique (Zhang et Suen) l'autre à critère métrique (Marthon). Mais avant cela nous avons à binariser notre image, ce qui a été fait en utilisant d'abord les résultats d'un détecteur de contours (Sobel) et cela afin d'obtenir une image binaire exploitable.

Les résultats de la squelettisation ont été convaincants, et on a opté pour l'algorithme de Marthon pour la suite de notre travail, vu que le squelette obtenu ne présente pas beaucoup de discontinuité.

Le chapitre suivant aura pour but l'étiquetage du squelette obtenu.

CHAPITRE V

**ETIQUETAGE DE
L'ARBRE CORONAIRE**

Chapitre V

ETIQUETAGE DE L'ARBRE CORONAIRE

V.1 INTRODUCTION

Nous récapitulerons brièvement dans cette introduction les difficultés liées à la tâche d'étiquetage de l'arbre coronaire. Tout d'abord, le réseau vasculaire coronaire est complexe, sa structure et sa géométrie varient avec les individus. Ensuite, les représentations d'arbres coronaires dont nous disposons, les angiographies par rayons X, sont difficiles à interpréter. Ce sont des projections 2D d'un objet initial 3D.

Par conséquent, les artères se croisent, se superposent et forment des boucles dans un plan de projection. Finalement, le résultat de la segmentation, tâche également complexe, est source d'inexactitudes dans le squelette extrait de l'angiographie (rupture ou perte de vaisseaux, ajout de structures non vasculaires).

Dans ce chapitre, nous introduirons quelques notions théoriques liées aux structures d'arbres pour l'interprétation du squelette artériel et des définitions sur le calcul de distance. L'état de l'art sur le sujet fait l'objet du premier paragraphe.

Dans la seconde partie de ce chapitre, nous détaillerons l'étiquetage de l'arbre coronaire ainsi que les différents algorithmes utilisés à cet effet.

V.2 DEFINITIONS

Un arbre est soit un arbre atomique (une *feuille*), soit un *noeud* et une suite de sous arbres. Graphiquement, un arbre est représenté comme suit :

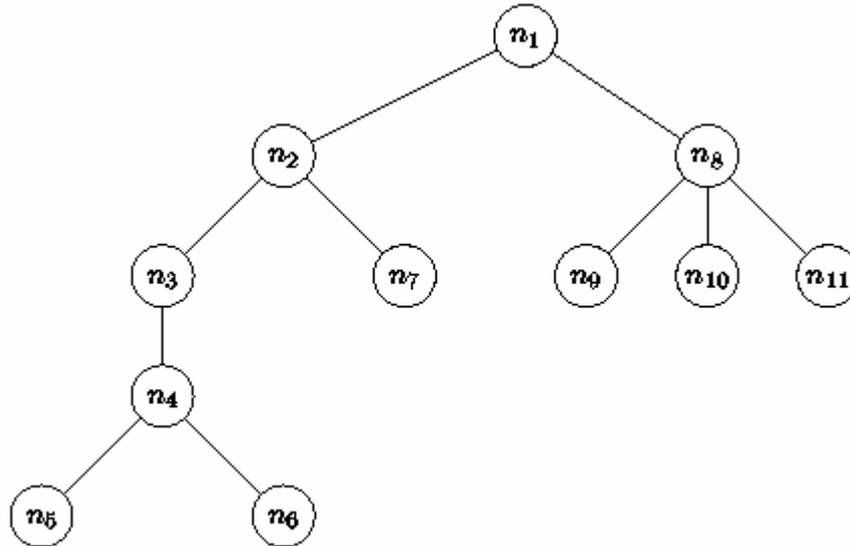


Figure V.1- Exemple d'arbre.

Le noeud n_1 est la *racine* de l'arbre, $n_5, n_6, n_7, n_9, n_{10}, n_{11}$ sont les *feuilles*, n_1, n_2, n_3, n_4, n_8 les *noeuds internes*. Plus généralement, l'ensemble des *noeuds* est constitué des noeuds internes et des feuilles.

V.2.1 Parcours d'arbre

Un algorithme de parcours dans un arbre est une procédure qui fournit en résultat la liste des noeuds de l'arbre sans oubli, ni répétition.

Il existe plusieurs moyens de parcourir les noeuds d'un arbre, les deux parcours les plus utilisés sont le parcours préfixé et le parcours postfixé. Ces parcours sont définis comme suit :

- Si un arbre A est vide, la liste vide constitue le parcours préfixé et postfixé de A .
- Si A est constitué d'un seul noeud, la liste composée de ce noeud constitue le parcours préfixé et postfixé de A .
- Si A est un arbre de racine n possédant k sous arbres : $n_1, n_2, n_3, \dots, n_k$:

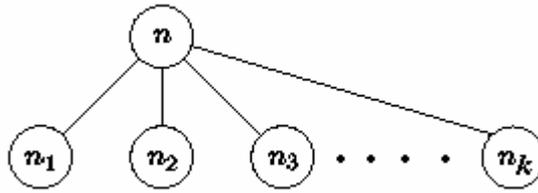


Figure V.2-Parcours de l'arbre.

- Pour effectuer le parcours préfixé des nœud de **A**, on commence par la racine n puis on entreprend le parcours préfixé des nœuds de n_1 , ceux de n_2 et ainsi de suite jusqu'aux nœud de n_k .

Exemple :

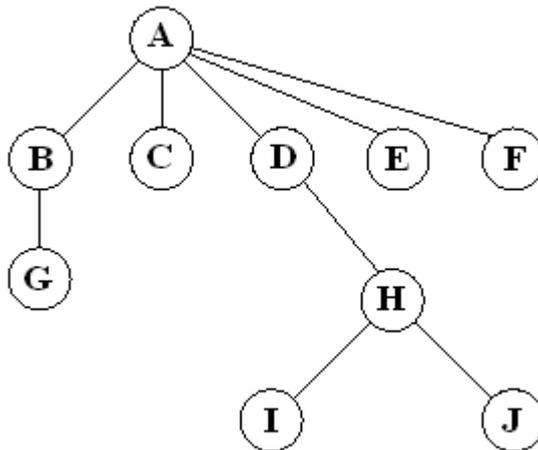


Figure V.3- Parcours préfixé de l'arbre.

Le résultat de parcours de l'arbre (**α**) en préfixé est : **A B G C D H I J E F**.

En étiquetant les nœuds par leurs rangs dans cet ordre de parcours, on obtient l'arbre suivant :

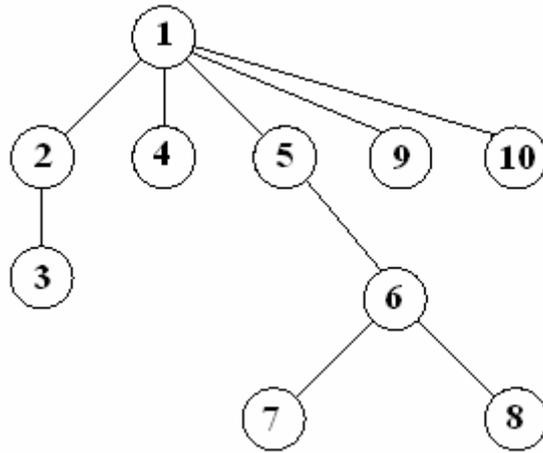


Figure V.4- Etiquetage de l'arbre dans l'ordre préfixé.

- Pour effectuer le parcours de l'arbre postfixé des nœuds de **A**, on effectue le parcours préfixé des nœuds de n_1 puis de n_2 et ainsi de suite jusqu'au nœud de n_k et on termine par la racine n .

Le résultat de parcours de l'arbre (**a**) en postfixé est : **G B C I J H D E F A**.

En étiquetant les nœuds par leurs rangs dans cet ordre de parcours, on obtient l'arbre suivant :

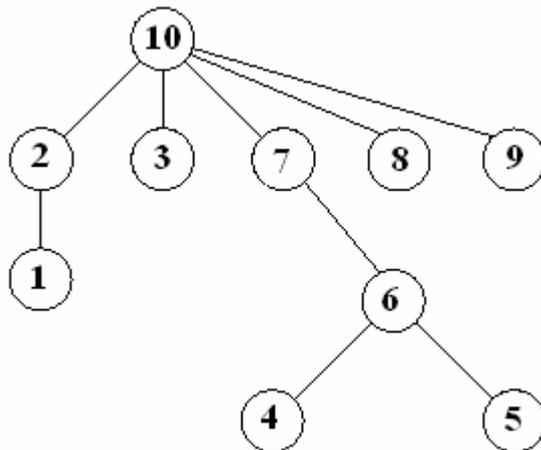


Figure V.4- Etiquetage de l'arbre dans l'ordre postfixé.

V.2.2 Définitions et notations

Définition 1

Un domaine d'arbre est un langage sur l'alphabet IN des entiers naturels où la concaténation est notée par le symbole « \cdot » dont l'élément neutre est 0.

On définit sur IN la relation « \leq » par :

$$a \leq b \Leftrightarrow \exists x \in IN / a \cdot x = b \text{ (c'est-à-dire } a \text{ est préfixe de } b \text{)}.$$

On définit d'autre part la relation :

$$a < b \Leftrightarrow a \leq b \text{ et } a \neq b$$

On dit que a est un ancêtre de b et b un descendant de a .

a et b sont dits incomparables s'ils sont tels que ni $a \leq b$, ni $b \leq a$.

Définition 2 :

$D \subset IN$ est un domaine d'arbre si et seulement si :

- i) D est fini.
- ii) $(a \in D \text{ et } a < b) \Rightarrow a \in D \cdot \Rightarrow b \in D$.
- iii) $a \cdot m \in D \Rightarrow a \cdot n \in D$ pour $1 \leq n \leq m$. Avec m et $n \in IN$

Définition 3

Un arbre étiqueté défini sur D est une application $\alpha : D \rightarrow \Sigma$.

Où Σ est un ensemble fini de symboles. $N(\alpha)$ Représente le nombre de nœuds dans α .

L'interprétation par domaine d'arbre permet d'attribuer de façon biunivoque un élément de D à chaque nœud de la structure d'arbre correspondante.

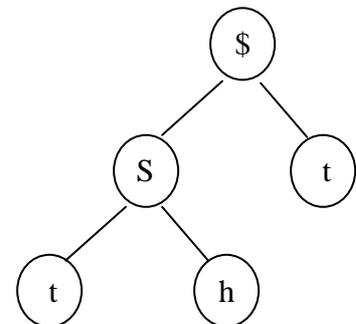
Le domaine d'arbre d'un arbre α est noté D_α

Exemple :

$$D = \{0, 1, 2, 1.1, 1.2\} , \Sigma = \{\$, S, t, h\}$$

$$\text{Si } \alpha(0) = \$$$

$$\alpha(1) = S$$



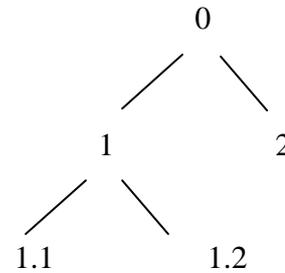
$$\alpha(2) = t$$

$$\alpha(1.1) = t$$

$$\alpha(1.2) = h$$

Alors l'arbre α est

D_α est :



$$N(\alpha) = 5$$

Définition 4

Un rang est une fonction $r : D_\alpha \rightarrow \mathbb{N}$ telle que $r(a) = n$, si le nœud a de α possède n fils.

Si a est une feuille alors $r(a) = 0$

Définition 5

On définit un opérateur $h_\alpha : D_\alpha \rightarrow \mathbb{N}$ qui ordonne une séquence de nœuds de D_α dans l'ordre postfixé.

Exemple :

Soit $D_\alpha = \{0, 1, 2, 1.1, 1.2\}$ alors

$$h_\alpha(1.1) = 1$$

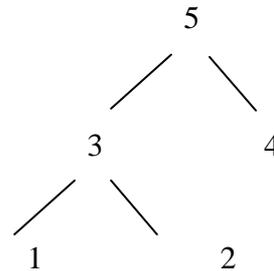
$$h_\alpha(1.2) = 2$$

$$h_\alpha(1) = 3$$

$$h_\alpha(2) = 4$$

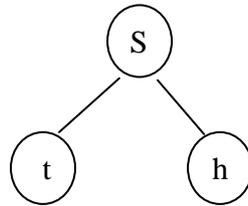
$$h_\alpha(0) = 5$$

En effet l'ordre postfixé de α est défini par :



Définition 6 :

Soit α un arbre étiqueté, et $a \in D_\alpha$. On note α/a le sous arbre de α dont la racine est a. Soit α l'arbre défini dans les exemples précédents alors $\alpha/1$ est :



V.3 PRINCIPE DE L'ETIQUETAGE

Poser une étiquette sur une image est essentiellement un problème de classification. Généralement, pour générer des étiquettes à partir d'une image, il faut générer pour chaque région une même étiquette (numéro) pour les distinguer des autres régions.

Dans notre étude, le principe de l'étiquetage est de déterminer la racine du squelette artériel et lui donner une étiquette, on continue le parcours avec la même étiquette tant qu'on n'a pas trouvé une bifurcation. Ensuite pour chaque nouvelle bifurcation on donne à l'artère secondaire une nouvelle étiquette, on répète cette opération jusqu'au parcours de tout le squelette artériel.

V.4 PARCOURS DE L'ARBRE

Après avoir extrait le squelette artériel, nous disposons d'une image formée d'un ou plusieurs amas de pixels allumés qui sont des ensembles connexes qu'il va falloir parcourir afin de le transformer (le coder) en arbre étiqueté binaire tel que chaque nœud de l'arbre porte pour étiquette une direction. La racine porte une étiquette spéciale.

Les directions choisies sont celles de l'alphabet de Freeman (Figure V-3). Une première étape consiste à détecter, par un balayage horizontal de l'image, le premier point du squelette artériel qui est le point le plus haut à gauche de l'image, ce point sera le sommet du squelette et par la même occasion le sommet de l'arbre qui va le coder.

Le premier point étant détecté, on cherche si l'un de ses 8-voisins est allumé, ce qui revient à faire une sorte de balayage circulaire autour du point considéré. Celui-ci est étiqueté comme ancien point et on se déplace vers le nouveau point. De proche en proche en parcourant entièrement l'amas de pixels qui est une artère.

- Si au voisinage d'un point aucun pixel n'est allumé, mis à part les anciens points, cela voudra dire qu'on a atteint l'extrémité de l'artère.
- Si au voisinage d'un point, il y a un pixel allumé, il s'agit d'un coin.
- Si au voisinage d'un point, il y a deux pixels allumés, il s'agit d'une intersection. Dans ce dernier cas, les coordonnées de l'intersection (les coordonnées du pixel courant et le nœud courant de l'arbre) sont mémorisées, on suit l'un des deux chemins jusqu'à son extrémité, on revient ensuite à l'intersection pour parcourir le deuxième chemin.

V.5 CODE DE FREEMAN

Lors du parcours du squelette artériel, chaque chemin parcouru est codé selon les primitives du code de Freeman. Cette méthode de codage consiste à coder chaque deux points successifs par un code allant de 0 à 7, correspondant à la plage angulaire à laquelle il appartient.

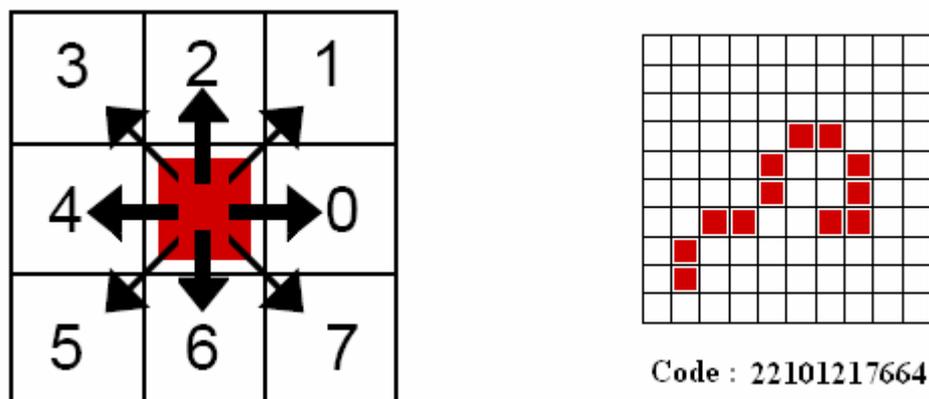


Figure V.5- Codage de Freeman

Nous avons utilisé dans notre algorithme le codage de Freeman pour parcourir le squelette artériel mais aussi pour l'étiquetage des différentes artères.

V.6 ALGORITHME D'ETIQUETAGE

Il se divise en trois parties :

- Un algorithme pour détecter la racine de l'arbre.
- Un algorithme pour détecter les points de bifurcation.
- Un algorithme qui prend en charge les étiquettes selon le codage de Freeman.

Algorithme pour détection de la racine :

```

Tant que ligne i de l'image < largeur
Tant que colonne j < longueur
  Si pixel [i,j] allumé alors
    Racine = [i,j]
  Sinon
    i++
    j++

```

Algorithme pour détecter les points de bifurcation :

```

Pour chaque pixel [i,j]
  Si nombre de voisin >1 alors
    [i,j] est un point de bifurcation
  Sinon
    Si nombre de voisins = 0
      [i,j] est un bout
    Sinon passer au point suivant

```

Algorithme d'étiquetage :

```

Pour chaque point de bifurcation
  Si voisin 1 est du code de Freeman alors
    Etiquette selon direction 1
  Sinon Etiquette selon direction du voisin 2

```

Nous avons implémenté ces algorithmes grâce au langage Builder C++, et appliqué sur un squelette « parfait », de largeur un pixel partout, et ne présentant aucune discontinuité. En suite nous avons appliqué le programme à notre squelette réel obtenu dans le chapitre précédent. La figure V.6 montre les résultats obtenus :

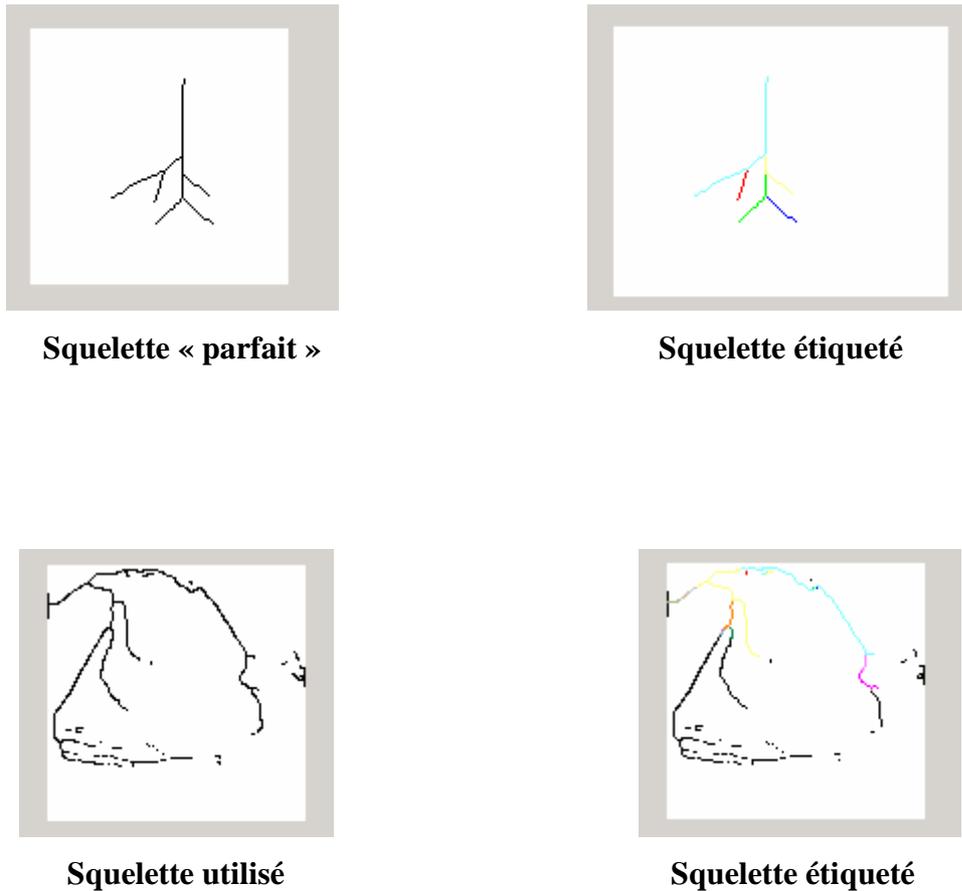


Figure V.6- Résultats de l'étiquetage

Nous remarquons que pour l'image réelle l'étiquetage ne s'est pas vraiment fait pour l'arbre entier, cela est dû à la qualité du squelette utilisé.

La solution consiste à étiqueter l'image par segment, c'est-à-dire prendre les régions où le squelette ne présente pas de discontinuité, et appliquer à chaque région notre algorithme.

V.7 CONCLUSION

Nous avons présenté à travers ce chapitre la méthode d'étiquetage utilisée, ainsi qu'une application avec le C++, les résultats dépendent de la qualité du squelette, en effet pour les zones où le squelette était bien déterminé, l'étiquetage s'est fait sans problème.

Dans la suite de notre travail, nous allons implémenter cet algorithme en utilisant le CCS, la première difficulté sera de manipuler l'image en tant que vecteur, et non plus en tant que matrice, la deuxième difficulté concerne l'affichage des couleurs pour les étiquettes, où nous avons opté pour des étiquettes à niveaux de gris.

CHAPITRE VI

ETUDE DE LA FAMILLE

C6000

Chapitre VI

ETUDE DE LA FAMILLE C6000

VI.1 INTRODUCTION

Un *Digital Signal Processor* ou *DSP* en anglais, soit « **processeur de signal numérique** », est un microprocesseur optimisé pour les calculs. Son application principale est le traitement du signal numérique (filtrage, extraction de signaux, etc.), d'où son nom.

Un DSP est un processeur dont l'architecture est optimisée pour effectuer des calculs complexes en un coup d'horloge, mais aussi pour accéder très facilement à un grand nombre d'entrées-sorties (numériques ou analogiques). La fonction principale utilisée dans le DSP est la fonction *MAC* (*Multiply and Accumulate*), c'est-à-dire une multiplication suivie d'une addition et d'un stockage du résultat (fonction très utilisée dans les calculs d'asservissement et de filtrage).

VI.2 ETUDE DE LA FAMILLE C6000

Le premier DSP appartenant à la famille C6000 que Texas Instrument a mis sur le marché fut le TMS320C6201 (C62x) ,c'était en 1997 . Il est à virgule fixe mais contrairement aux autres DSP à virgule fixe des générations précédentes (C1x,C2x et C5x) le C62x est basé sur une architecture VLIW (Very Long Instruction Word) qui présente des instructions plus simples.

Un processeur à virgule fixe est plus efficace quand il s'agit d'applications qui requièrent des batteries (téléphone cellulaire..) vu leur faible consommation d'énergie. [11]

VI.2.1 ARCHITECTURE DU C6000

Le TMS320C6211 ,mis en œuvre sur une carte DSK est un processeur à virgule fixe, utilisant une architecture VLIW(Very Long Instruction Word) lui procurant de grands avantages pour les applications multicanaux et multitâches.[18]

Le chemin de données (bus internes) est formé d'un bus d'adresse programme de 32bits, d'un bus programme de 256bits pour faire loger huit instructions de 32bits, deux bus d'adresse de données 32bits, deux bus de données 32bits et deux bus de données store de 64bits. Avec un bus de 32bits pour les adresses, l'espace mémoire total adressable est de $2^{32}=4\text{GB}$ incluant quatre espaces mémoires externes : CE0, CE1, CE2 et CE3. La figure III.1 montre le diagramme en blocs des parties fonctionnelles du C6211.

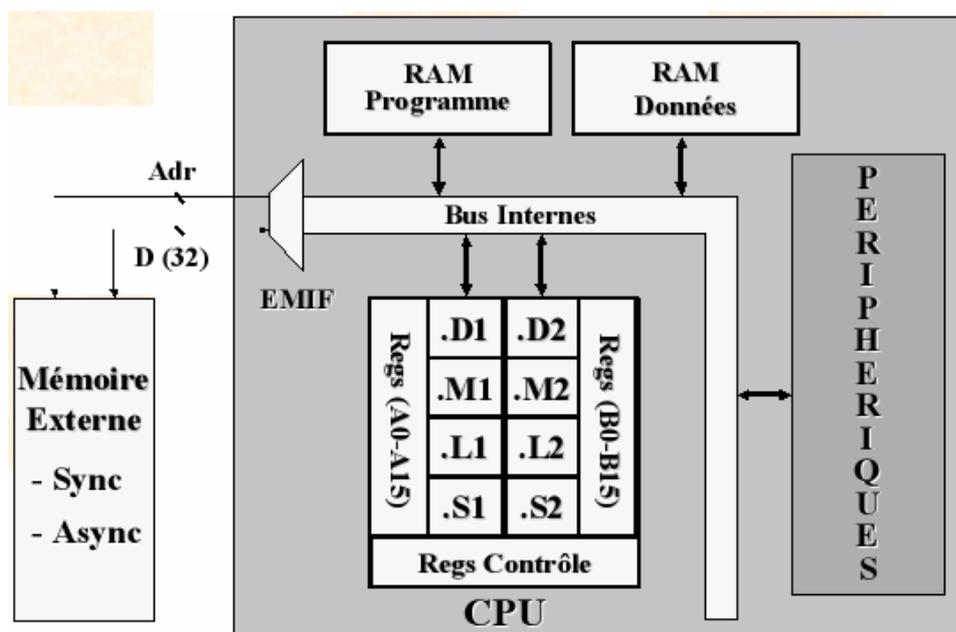


Figure VI.1- Block Diagram du TMS320C6211

L'indépendance des bancs mémoires des processeurs C6x permet un double accès en mémoire en un seul cycle d'instruction, et donc deux instructions load ou store peuvent être exécutées en parallèle tant que les données à y accéder résident dans des blocs différents.

Le C6211 du DSK inclut 72kB de mémoire interne adressable à partir de l'adresse 0x00000000, et 16MB de SDRAM adressable via CE0 à partir de l'adresse 0x80000000. Le DSK inclut aussi une mémoire flash de 128kB à partir de 0x90000000.

Le tableau suivant donnera les différentes caractéristiques de la famille C6x

Device	MIPS	MHz	Kbytes	pins	mm	W	\$	Periphs
6201B	1600	200	128	352	27	1.9	80-110	D2H
6202	2000	250	384	352	27	1.9	120-150	D3X
6211	1200	150	72	256	27	1.5	20-40	E2H

TMS320	MFLOPS	MHz	Kbytes	pins	mm	W	\$	Periphs
6701	1000	167	128	352	35	1.9	170-200	D2H
6711	600	100	72	256	27	0.9	20-40	E2H

Peripherals Legend:**D,E:** DMA,EDMA**2,3:** # of McBSPs**H,X:** HPI, XBUS**Tableau VI.1** caractéristiques de la famille C6x**VI.2.2 LE CPU**

Le CPU consiste en deux chemins de données A et B, chaque chemin comporte quatre unités fonctionnelles indépendantes :

- Une unité pour la multiplication (.M).
- Une unité pour les opérations logiques et arithmétiques (.L).
- Une unité pour les branchements, la manipulation des bits et opérations arithmétiques (.S)
- Une unité pour le loading/storing et opérations arithmétiques (.D).

Les unités .S et .L sont utilisées par les instructions arithmétiques, logiques, et de branchements. Tous les transferts de données se font via les unités .D.

Les opérations arithmétiques telles que celles d'addition ou de soustraction (ADD ou SUB), peuvent être exécutées par toutes les unités exceptées les unités .M. Les huit unités fonctionnelles consistent en quatre unités arithmétiques et logiques ALUs travaillant en virgule fixe (deux .L et deux .S), deux autres unités arithmétiques virgule fixe (.D), et deux multiplieurs virgule fixe (.M). Chaque unité fonctionnelle peut lire directement du ou écrire directement dans un registre du fichier à travers son propre chemin. Chaque chemin inclut un ensemble de seize registres de 32bits, de A0 à A15 et de B0 à B15. Donc il y a 32 registres à usage général; certains sont réservés pour les modes d'adressage spéciaux ou utilisés par les instructions conditionnelles.

Deux chemins croisés (cross-paths 1x et 2x) permettant aux U.F, à partir d'un chemin de données, l'accès à un opérande de 32bits dans un fichier du registre du banc opposé. Un maximum de deux accès par cycle utilisant le cross-path est permis.

VI-2-3 LES TIMERS

Grâce à deux timers de 32bits, on peut cadencer les périphériques, compter les évènements, générer des impulsions de commande, interrompre le CPU, et déclencher le transfert à travers le DMA.

VI.2.4 LES REGISTRES

Deux fichiers contenant chacun un ensemble de registres (A0-A16 et B0-B16). Les registres A0, A1, B0, B1 et B2 sont utilisés comme étant des registres conditionnels. Les registres de A4 à A7 et de B4 à B7 utilisés par l'adressage circulaire. Les registres de A0 à A9 et de B0 à B9 (excepté le B3) sont temporairement des registres à usage général. Les registres de A0 à A15 et de B10 à B15 utilisés sont enregistrés et après restaurés avant le retour d'un sous programme. [19]

Une donnée sur 40bits peut être rangée dans une paire de registres. Les 32bits de poids faible sont stockés dans les registres à indice pair (c.à.d A2) et les 8bits qui restent sont stockés dans les 8LSBs du registre impair suivant (c.à.d A3).

Ces 32 registres sont considérés comme des registres à usage général. Plusieurs registres spéciaux existent aussi pour le contrôle et les interruptions; par exemple : le registre de mode d'adressage (AMR) utilisé pour l'adressage circulaire et les registres du contrôle d'interruptions.

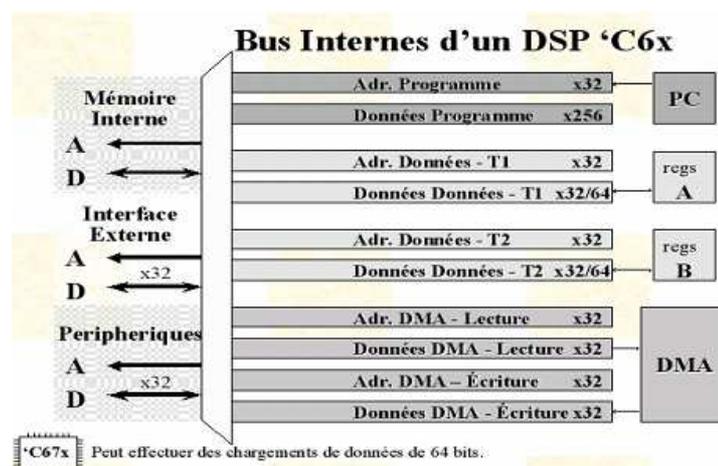


Figure VI.2- Bus internes.

VI-2-5 LES MsBSP

Deux multichannels buffered serial ports (McBSP) sont disponibles servant d'interfaces pour les périphériques standard. Ils présentent des avantages tels que une communication en mode full-duplex, une synchronisation et datagrammes indépendants pour l'émission et la réception, ainsi qu'une interface directe avec le AC97 et dispositifs conformes d'IIS. Ils permettent diverses tailles de données entre 8 et 32 bits.

La communication de données externe peut se produire pendant que des données sont déplacées intérieurement. La figure 3.8 montre un schéma fonctionnel interne d'un McBSP.

Les broches transmission de données (DX) et réception de données (DR) sont employées pour la communication de données. Le contrôle des paramètres (horloge et synchronisation des trames) s'effectue via CLKX, CLKR, FSX, et FSR. Le CPU ou le contrôleur DMA lit les données à partir du registre de réception de donnée (DRR) et écrit les données à transmettre dans le registre de transmission de données (DXR). Le registre à décalage de transmission (XSR) envoie les données à travers DX, alors que le registre à décalage de réception (RSR) copie les données reçues sur le DR dans le buffer du registre de réception (RBR). Les données dans RBR sont transférées par la suite à DRR et lues par l'unité centrale de traitement ou le contrôleur DMA. On trouve d'autres registres comme le registre de contrôle du port série (SPCR), le registre de contrôle d'émission/réception (XCR/RCR), le registre d'activation des canaux d'émission/réception (XCER/RCER) et le registre de contrôle de broche (PCR). Les deux McBSP sont utilisés pour l'entrée et la sortie à travers le codec du DSK. McBSP0 est utilisé pour le contrôle et McBSP1 pour l'émission et la réception des données.

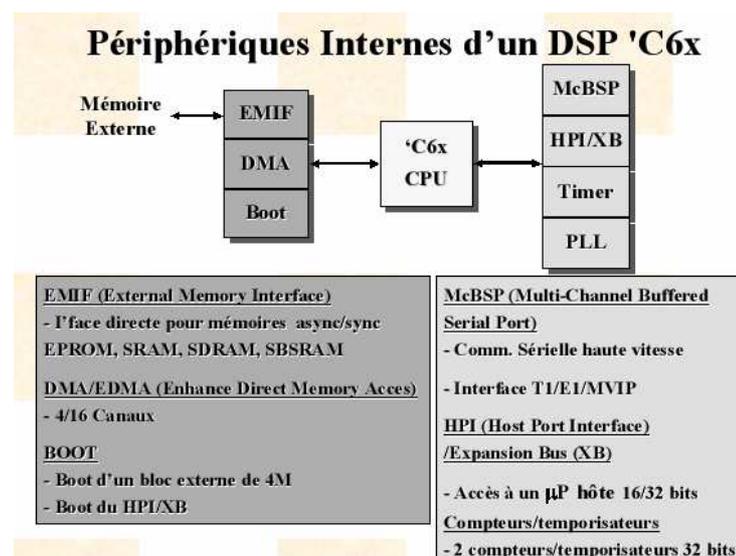


Figure VI.3 Périphériques internes

VI.3 OUTILS DE DEVELOPPEMENT

TI a mis au point plusieurs outils de développement permettant entre autre fonctions la génération de codes, le développement d'algorithmes ainsi que le contrôle de leur exécution et la gestion du software et hardware.

On trouve pour le développement :

- Software : Le Code Composer Studio, le compilateur C/C++/ et l'Assembleur, ainsi que le Real Time Data eXchange (RTDX) et DSP/BIOS pour le transfert de données entre un ordinateur hôte et les périphériques du DSP.
- Hardware : On trouvera dans cette catégorie, l'émulateur XDS (Extended Development System) et EVM (Evaluation Module)

VI.3.1 CODE COMPOSER STUDIO

Code Composer Studio (CCS) est un environnement intégré de développement de code pour les DSP de Texas Instrument. Il est fourni en standard avec la carte de développement pour le DSP.

CCS fournit plusieurs outils pour faciliter la construction et la mise au point des programmes de DSP. Il comprend un éditeur de code source, un compilateur de langage C/C++, un assembleur de code source, un éditeur de liens et un environnement d'exécution qui permet de télécharger un programme exécutable sur une carte cible, de l'exécuter et de le déboguer au besoin. CCS comprend aussi des outils qui permettent l'analyse en temps réel d'un programme en cours d'exécution et des résultats produits. Finalement, il fournit un environnement de gestion de fichiers qui facilite la construction et la mise au point des programmes.



VI.3.1.1 GENERATION DE CODES

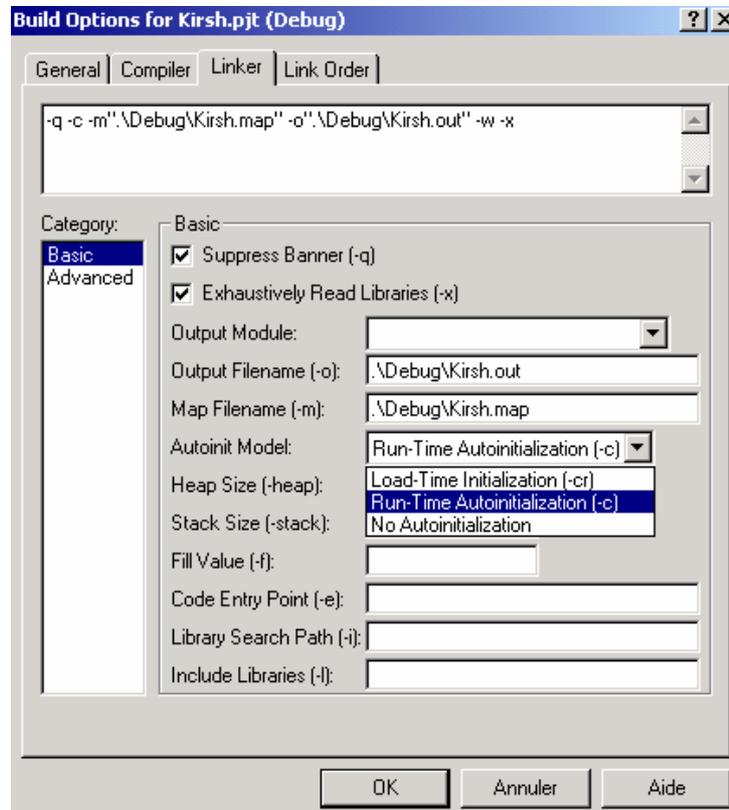
Le code composer (CCStudio IDE) fournit une interface graphique adéquate pour l'utilisation des outils de génération de code (code generation tools). Il garde en permanence les informations utilisées pour construire et compiler un programme ou une librairie

La figure VI-6 montre les différentes étapes du développement software d'un code exécutable .

Un fichier projet, dont nous allons voir sa création plus loin, enregistre :

- les noms de fichiers du code source et la librairie des projets ;
- des options de compilation, d'assemblage et de lien ;
- il inclut aussi les dépendances des fichiers.

Les options de compilation, d'assemblage et du linker peuvent être spécifiées et configurées via la boîte de dialogue « Build options », figure VI-5. On peut avoir l'accès à presque toutes les options. Celles qui ne sont pas représentées peuvent être introduites aisément à travers la ligne de commande.



FigureVI.4- Build Options

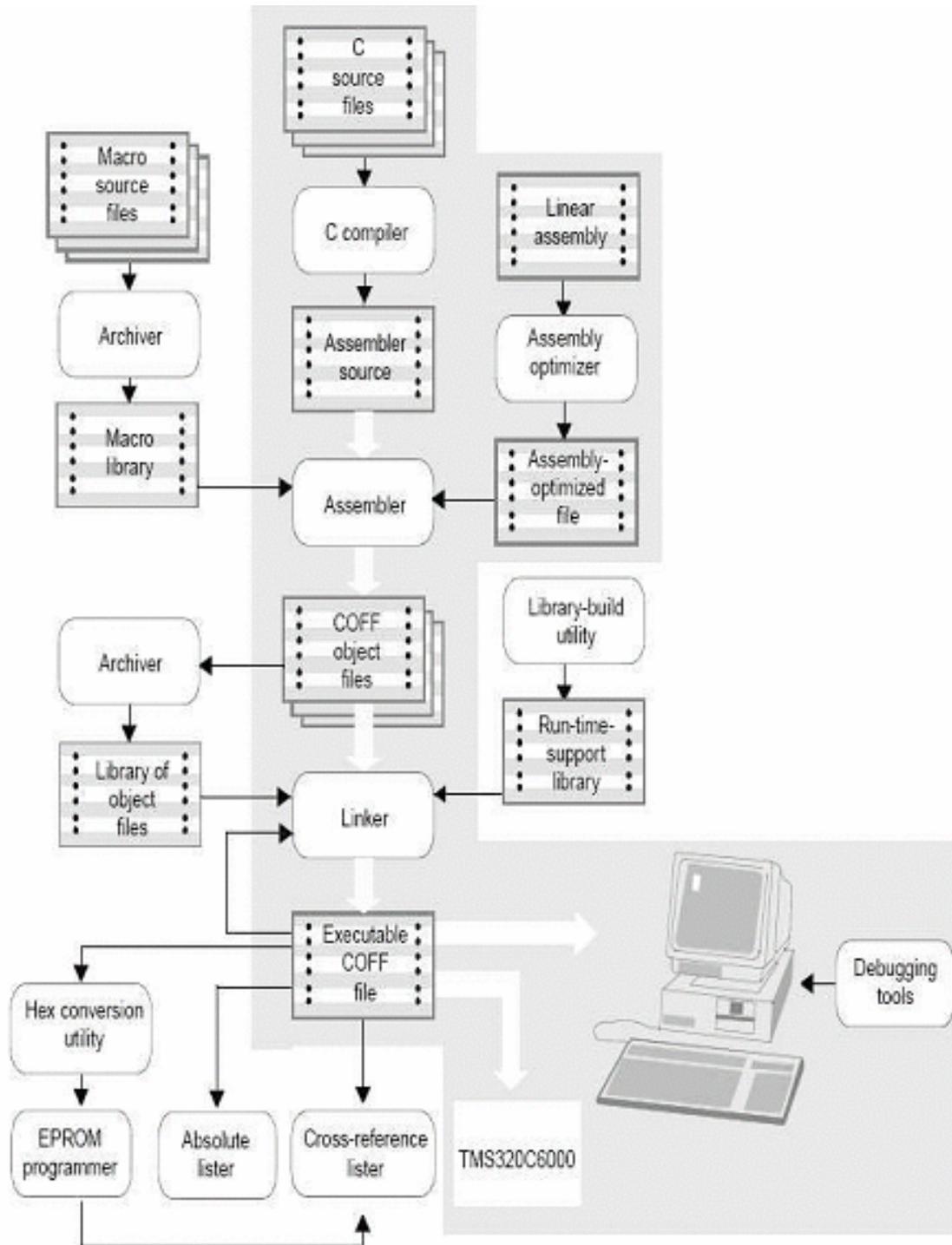


Figure VI.5- Processus du développement du code exécutable.

VI.3.1.1.1 Outils de développement pour l'assembleur

Les outils de développement de langage d'assemblage sont les suivants : [20]

- **Assembleur** : l'assembleur traduit les fichiers source, écrits en langage assembleur, en fichiers objet de langage machine. Le langage machine est basé sur le format de fichier « common object file format » (COFF).
- **Archiver** : l'archiver permet de rassembler un groupe de fichiers dans un seul fichier d'archives simple appelé une bibliothèque. De plus, il permet de modifier une bibliothèque en supprimant, remplaçant, extrayant, ou ajoutant des membres (routines). Une des applications les plus utiles de l'archiver est de construire une bibliothèque des modules d'objet.
- **Éditeur de liens (Linker)** : l'éditeur de liens combine les fichiers objet dans un fichier objet exécutable par la cible. Pendant la création du fichier exécutable, il exécute la mise en mémoire adressable disponible, selon la cible, des fonctions définies par le fichier de commande et associe les références externes.
- **Absolute Lister** : il accepte les fichiers exécutables comme entrée et crée des fichiers .abs en sortie. En assemblant ce fichier on obtient une liste du code contenant les adresses absolues des instructions du code exécutable.
- **Cross-reference Lister** : le générateur de liste de correspondance emploie les fichiers exécutables pour produire une liste de correspondance montrant des symboles, leurs définitions et leurs références dans les fichiers source liés. C'est donc un rapport de l'opération exécutée par le linker.
- **Utilitaire de conversion hexadécimal** : son utilité réside dans la conversion de fichier exécutable de format COFF en format adapté aux EPROM Ti-Étiqueté, d'ASCII-hex, d'Intel, de Motorola-S, ou de Tektronix. L'EPROM en question fait l'objet d'une mémoire de stockage du code exécutable via laquelle le DSP est amorcé.

III-3-1-1-2 Outils de développement pour C/C++

Les outils de développement pour C/C++ sont :

Compilateur C/C++ : le compilateur de C/C++ accepte le code source C/C++ et produit le code source en langage assembleur. Donc il s'agit de compiler et faire l'assemblage des fichiers ; si une optimisation est invoquée l'optimisateur de code C intervient.

Optimisateur du code Assembleur : il permet de prendre en considération le code écrit en assembleur linéaire .sa (Linear assembler) c'est-à-dire qu'il est écrit sans tenir compte

du parallélisme ou une optimisation d'utilisation des registre. Son rôle est d'optimiser cette structure en employant les registres appropriés, ainsi qu'un autre apport d'optimisation pour les boucles, et cela pour obtenir un code fortement parallélisé.

Utilitaire de construction de bibliothèques : utilisé pour réaliser un archivage de routines fréquemment utilisées sous forme de bibliothèques, cet utilitaire génère un fichier d'extension .lib. Des bibliothèques sont disponibles sont fournies par le concepteur du logiciel CCS (TI), telle que la bibliothèque rts6200.lib utilisée par le linker afin de construire le fichier exécutable adéquat à une cible de la famille C62x.

code source en langage assembleur. Donc il s'agit de compiler et faire l'assemblage des fichiers ; si une optimisation est invoquée l'optimisateur de code C intervient. Ces fonctions d'optimisation sont définies dans la boîte de dialogue « Build Options ».

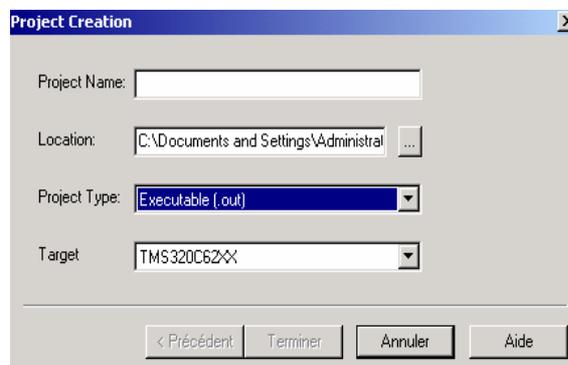
-Optimisateur du code Assembleur : il permet de prendre en considération le code écrit en assembleur linéaire .sa (Linear assembler) c'est-à-dire qu'il est écrit sans tenir compte du parallélisme ou une optimisation d'utilisation des registre. Son rôle est d'optimiser cette structure en employant les registres appropriés, ainsi qu'un autre apport d'optimisation pour les boucles, et cela pour obtenir un code fortement parallélisé.

- Utilitaire de construction de bibliothèques : utilisé pour réaliser un archivage de routines fréquemment utilisées sous forme de bibliothèques, cet utilitaire génère un fichier d'extension .lib. Des bibliothèques sont disponibles sont fournies par le concepteur du logiciel CCS (TI), telle que la bibliothèque rts6200.lib utilisée par le linker afin de construire le fichier exécutable adéquat à une cible de la famille C62x.

III-3-1-1-3 Création d'un projet :

Avant de commencer un développement il faut créer un projet ,la démarche est la suivante :

Il faut tout d'abord créer le fichier de gestion du projet (extension .pjt), on va dans le fichier Project et on clique sur new, la fenêtre suivante va apparaître (Fig VI.8)



FigureVI.6- Création d'un nouveau projet.

Ensuite on peut ajouter des fichiers source à ce projet, comme par exemple des fichiers C, fichiers d'entête (math.h, stdio.h, etc.), le fichier donnant la tables des vecteurs d'interruption, et les librairies (.lib) ainsi que le fichier commande (.cmd), figure VI-8

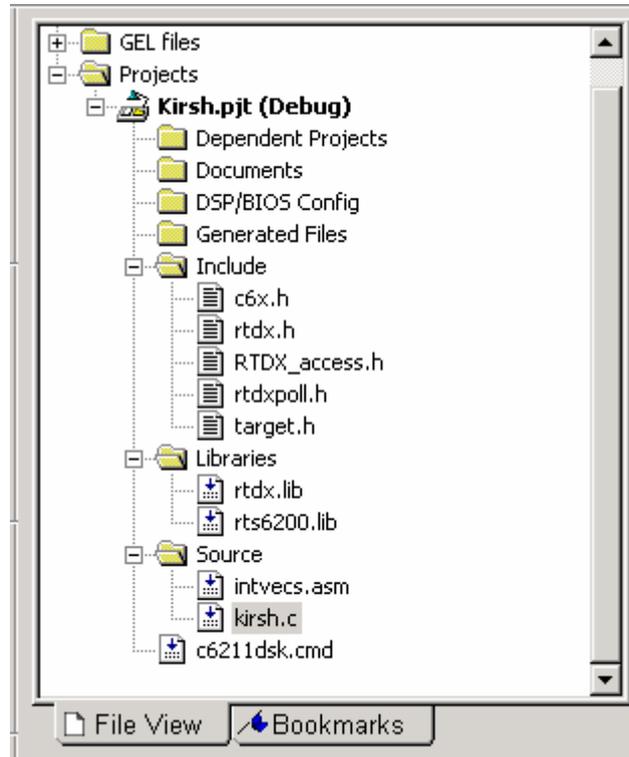


Figure VI.7- Différent fichiers composant un projet.

Le fichier .cmd est un fichier de commande dans lequel est définie la répartition en sections, dans la mémoire, des différents éléments d'une application, telles que les sections pour les variables locales et globales, et celle concernant le code.

Une fois que le fichier code source principal a été écrit, et que tous les fichiers annexes sont présents dans le projet, il faut " construire " le projet, c'est à dire générer un fichier exécutable compréhensible par le DSP. Pour cela, Il faut utiliser l'instruction " Rebuild all ", dans le menu " project ". Cette fonction réalise toutes les opérations nécessaires à la génération du fichier compréhensible par le DSP, c'est à dire sauvegarde des fichiers, assemblage des fonctions écrites en langage assembleur, élaboration des liens entre les différents fichiers, et compilation du source C principal. Une fois le fichier généré, s'il n'y a pas d'erreur, on obtient un message signalant l'absence d'erreurs. On peut alors lancer le programme sur le DSP, à l'aide de la commande " run " du menu " debug ". On peut stopper l'exécution du programme à l'aide de la fonction " halt ", et effacer le programme de la mémoire du DSP, à l'aide de la commande " reset DSP ".

VI.3.2 CARTE DSK

La carte utilisée pour le développement est formée des composants suivants :

- DSP TMS320C6211 à 150MHz.
- 16 M Bytes de mémoire externe SDRAM 100MHz.
- 128 K Bytes de mémoire FLASH programmable et effaçable.
- Port parallèle d'interfaçage (SPP) entre la carte DSK et le PC hôte.
- Port d'interfaçage hôte permet l'accès à toutes les ressources mémoire du DSP à travers le port parallèle (moyen d'implémentation du code).
- Outil embarqué JTAG (Joint Test Action Group), accès en direct par le support XDS510 ou bien par émulation via le port parallèle.
- 16-bit audio codec.
- Quatre LED d'indication utilisées lors de l'amorçage.
- Deux supports d'extension de mémoire ou périphérique.

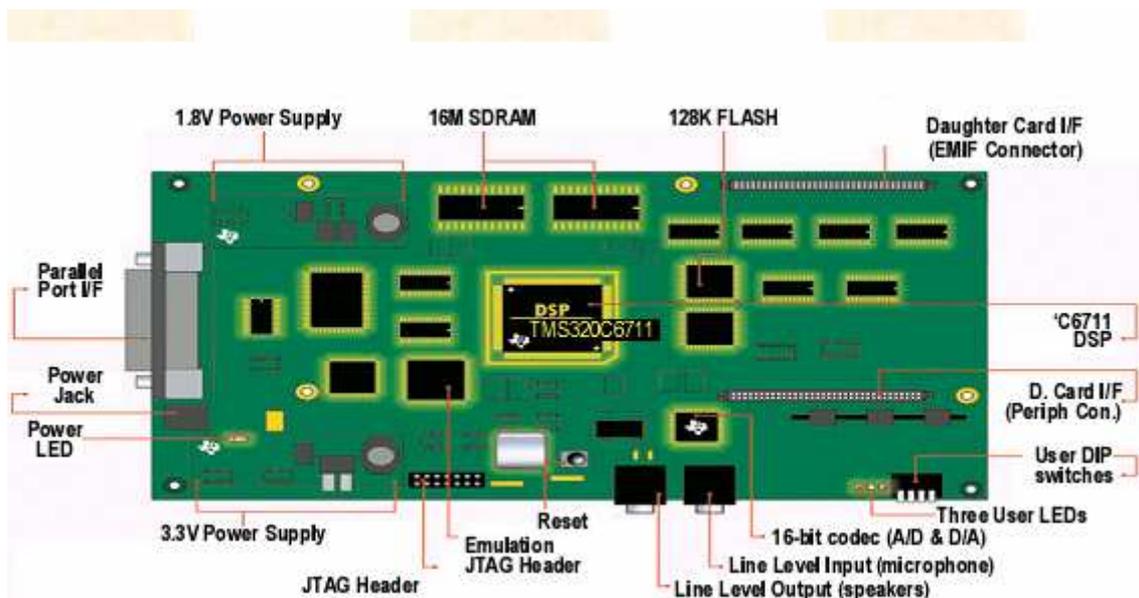


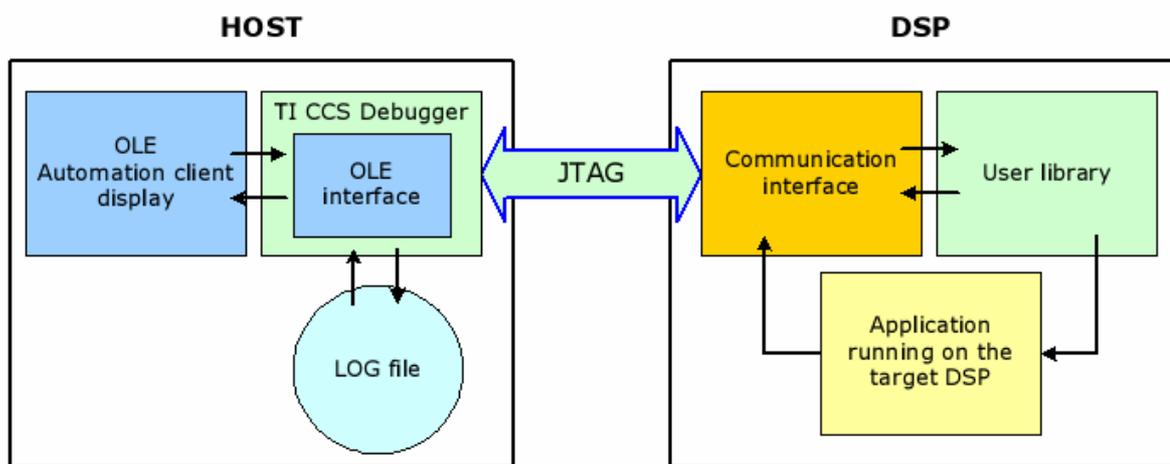
Figure VI.8- Carte DSK.

VI.3.3 LE RTDX

Le Real Time Data eXchange (RTDX) permet aux développeurs de systèmes de transférer des données entre un ordinateur hôte et les périphériques de la cible. Il s'agit d'une communication bidirectionnelle qui est en mode half-duplex lorsqu'on utilise le port parallèle. Les données reçues de la cible peuvent être analysées et visualisées sur l'ordinateur hôte, ainsi qu'une possibilité d'ajuster les paramètres de l'application en utilisant les outils disponibles sous CCS.

Le RTDX se compose des deux composants de cible et du PC hôte. Une petite bibliothèque de RTDX fonctionne sur l'application de la cible. L'application de cible fait des appels de fonction à l'api (interface de programmation d'application) de cette bibliothèque afin de transférer des données vers ou à partir de la cible, par l'intermédiaire de l'interface JTAG (figure VI.7).

Sur la plateforme hôte, une bibliothèque de RTDX est disponible fonctionnant en conjonction avec CCS IDE. Pour visualiser les données on peut utiliser une interface graphique basée sur les API RTDX et programmée sous Visual C++ ou Visual Basic ou bien le LabView de National Instruments.



FigureVI.9- Connexion entre hôte et cible via JTAG.

VI.3.3.1 La communication PC-Cible :

Dans la communication PC-Cible, le canal d'entrée doit être configuré.

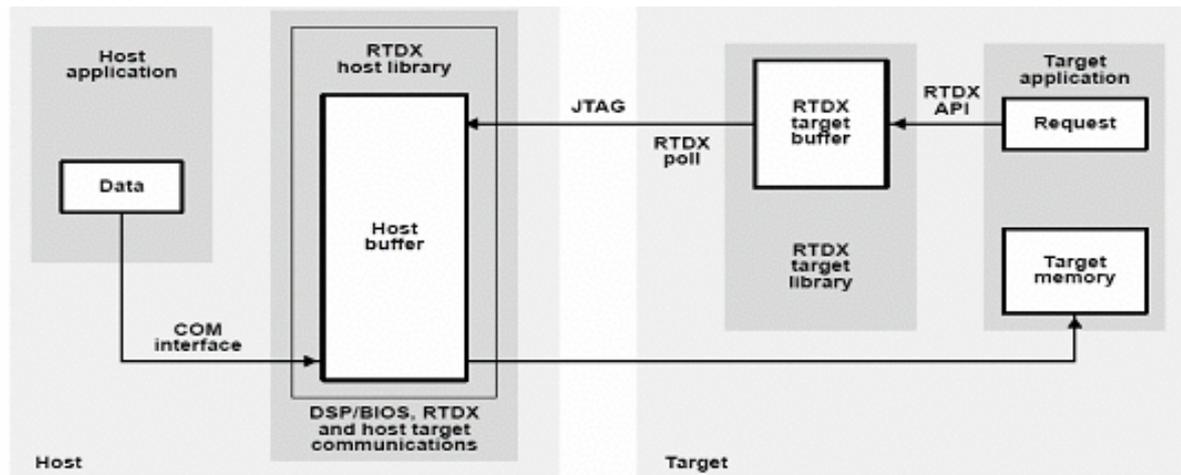


Figure VI.10- Etablissement de la connection PC-Cible.

La cible envoie une demande de réception de données en utilisant les routines définies lors de la programmation. Cette demande est enregistrée dans un buffer de la cible et envoyée par la suite au PC via l'interface JTAG (ou port parallèle). Une fois la demande reçue, le PC hôte écrit les données dans le buffer défini par la librairie du PC via l'interface COM (liaison logique entre l'application client et la librairie RTDX). Le PC ou mieux encore l'environnement de développement peut procéder au transfert des données via l'interface JTAG à l'adresse mémoire désirée au préalable (Figure VI.11.)

VI.3.3.2 Connection Cible-PC :

Dans la communication Cible-PC hôte, le canal de sortie doit être configuré. Les données sont écrites dans le canal de sortie moyennant les routines définies dans l'environnement de développement. Ces données sont immédiatement enregistrées dans un buffer au niveau de la cible définie dans une librairie (créée par compilation du projet). Les données présentes dans le buffer sont alors envoyées au PC hôte via l'interface JTAG (ou bien le port parallèle). La librairie RTDX générée pour le PC hôte reçoit ces données de l'interface JTAG et les enregistre dans un buffer mémoire ou un fichier (figure VI.12).

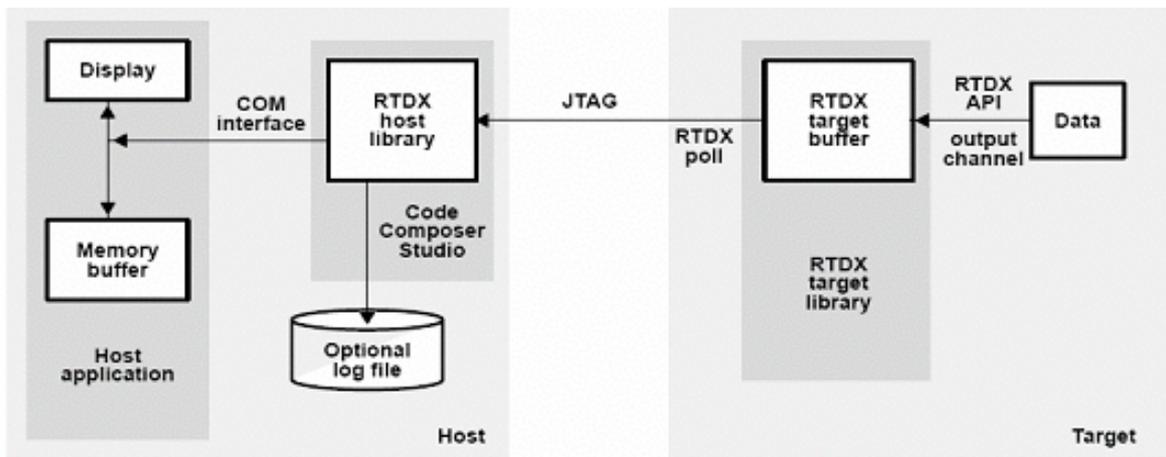


Figure VI.11- Connection Cible-PC.

VI.4 CONCLUSION

Nous avons fait une étude sommaire du DSP C6000 de TI, et nous avons pu voir certaines améliorations apportées aux DSP C6000 par rapport à ses prédécesseurs, sur le plan architectural :

- ✓ Architecture VILOCITI
- ✓ Interfaces de communication

et de programmation :

- ✓ performances du compilateur C
- ✓ options d'environnement

C'est une cible qui convient très bien lorsqu'il s'agit d'un traitement d'image en temps réel

CHAPITRE VII

**SIMULATION SUR DSP
C6000**

CHAPITRE VII

SIMULATION SUR DSP C6000

VII.1 INTRODUCTION

Dans ce chapitre nous allons détailler les différentes étapes pour la simulation des algorithmes utilisés sur le simulateur du C6211, le programme d'étiquetage sera détaillé afin d'expliquer l'exécution d'un programme sous CCS.

VII.2 CONFIGURATION DU CCS

Le CCS Setup, nous permet de configurer le logiciel CCS IDE afin de travailler sur différents matériels ou cibles du simulateur. En effet, le CCS Setup dispose de plusieurs configurations pour plusieurs familles et une configuration du système est exigée avant de créer une quelconque application, afin de déterminer quels outils seront utilisés par le CCS.

Dès l'installation du CCS, une configuration du simulateur C62x est fournie par défaut.

Pour créer une nouvelle configuration du système on suit les étapes suivantes :

- **Etape 1** : On ouvre le CCS Setup en double cliquant sur l'icône du CCS Setup localisée sur le bureau.
- **Etape 2** : On enlève l'ancienne configuration (ou celle installée par défaut) en cliquant sur le bouton « Clear » (effacer) de la boîte de dialogue « Import Configuration ».
- **Etape 3** : On clique ensuite sur le bouton « Yes » pour confirmer la commande « Clear ».
- **Etape 4** : On sélectionne la configuration standard qu'on veut installer sur la liste « Available Configuration »,
- **Etape 5** : On clique sur le bouton « Import » pour importer notre configuration sélectionnée. Si cette dernière a plus d'une cible on répète les étapes 4 et 5 jusqu'à ce que nous sélectionnons une configuration pour chaque cible.

- **Etape 6** : On clique sur le bouton « Save and Quit » pour sauvegarder la nouvelle configuration.
- **Etape 7** : On clique sur le bouton « Yes » quand le message suivant apparaît : « Start Code Composer studio on exit ». Le CCS Setup sera fermé et le CCS IDEs'ouvre automatiquement utilisant la nouvelle configuration créée.

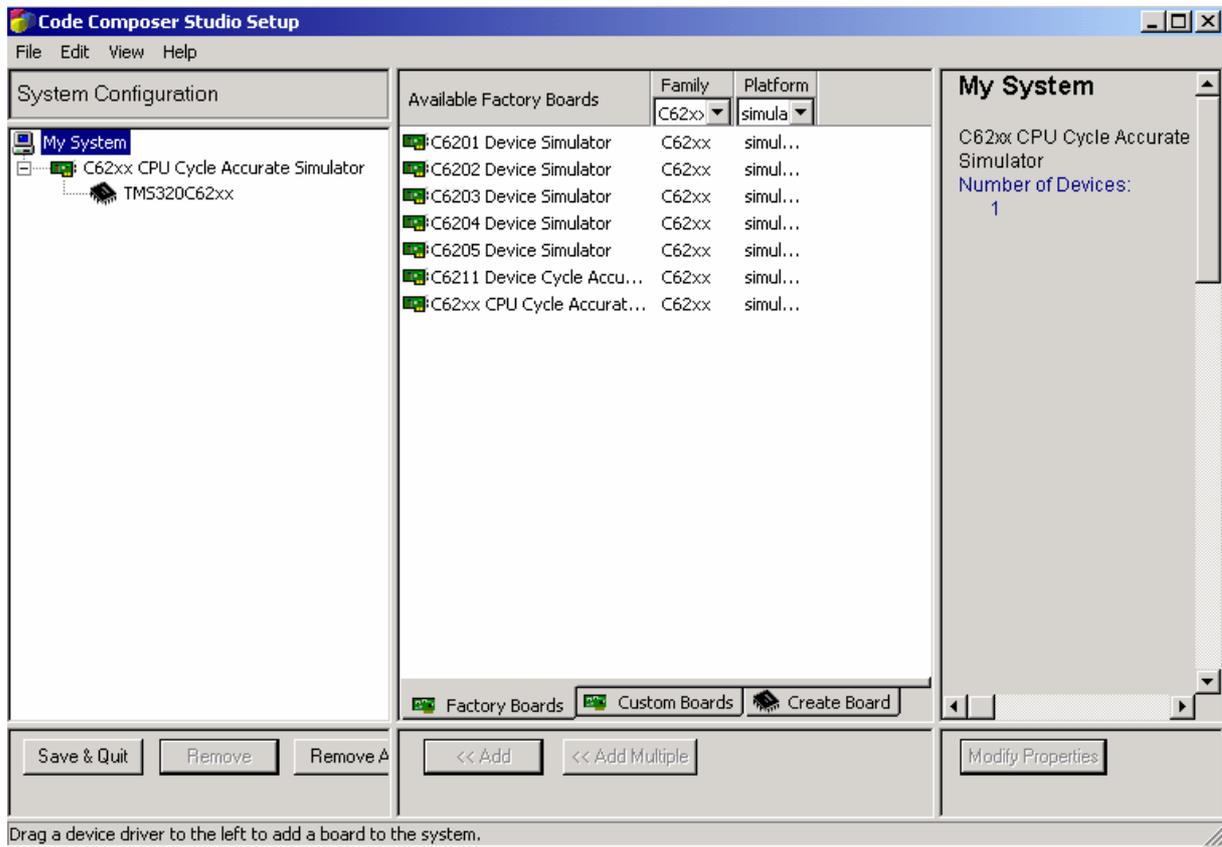


Figure VII.1 Configuration du CCS

VII.3 GESTION D'UN PROJET

VII.3.1 CREATION D'UN NOUVEAU PROJET

Les procédures suivantes nous permettent de créer de nouveaux projets, un ou plusieurs à la fois, avec un nom unique pour chaque projet.

L'information pour un projet est enregistrée dans un fichier à extension (*.pjt).

Note : On peut ouvrir plusieurs projets en même temps.

Les étapes de création d'un projet sont:

- **Etape 1** : Sur le menu du CCS IDE on sélectionne Projet → New, la fenêtre ci-dessous apparaît :

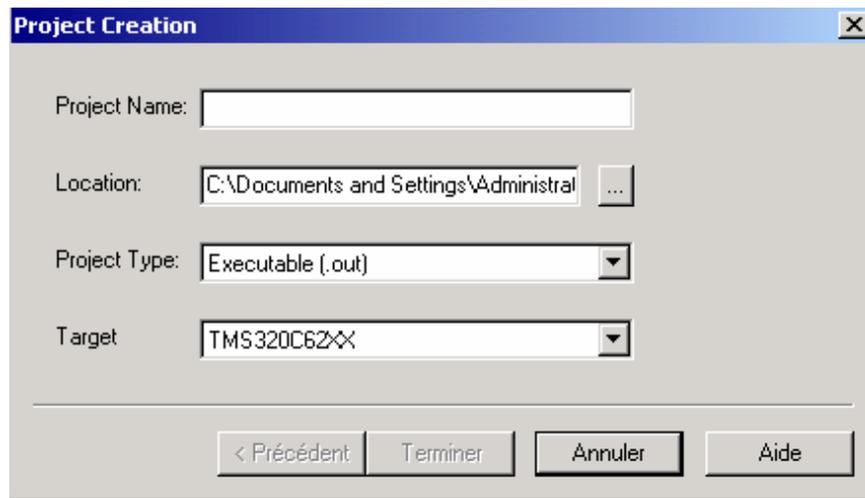


Figure VII.2 Boite de dialogue nouveau projet

- **Etape 2** : Sur le champ « Project Name » on écrit le nom qu'on veut attribuer à notre projet.
- **Etape 3** : Sur le champ « Location » on choisit le chemin où on veut enregistrer notre projet.
- **Etape 4** : Sur le champ « Project Type » on choisit le type du projet dans la liste proposée, telle que : Exécutable (.out) pour que le projet produise un dossier exécutable ou Bibliothèque (.lib) pour construire une bibliothèque de l'objet.
- **Etape 5** : sur le champ « Target » on choisit la famille qui identifie notre CPU. Cette information s'avère très nécessaire lorsque les outils sont installés pour plusieurs cibles.
- **Etape 6** : on clique sur « Finish », le CCS IDE crée le dossier de notre projet (projectname.pjt) qui contiendra tous les dossiers utiles pour ce dernier. Le nouveau projet devient automatiquement le projet actif et aura les options de débogage et de liens (linker) fournies par TI.

Après avoir créé notre projet on ajoute à sa liste : le fichier code source, bibliothèque et le fichier de commande.

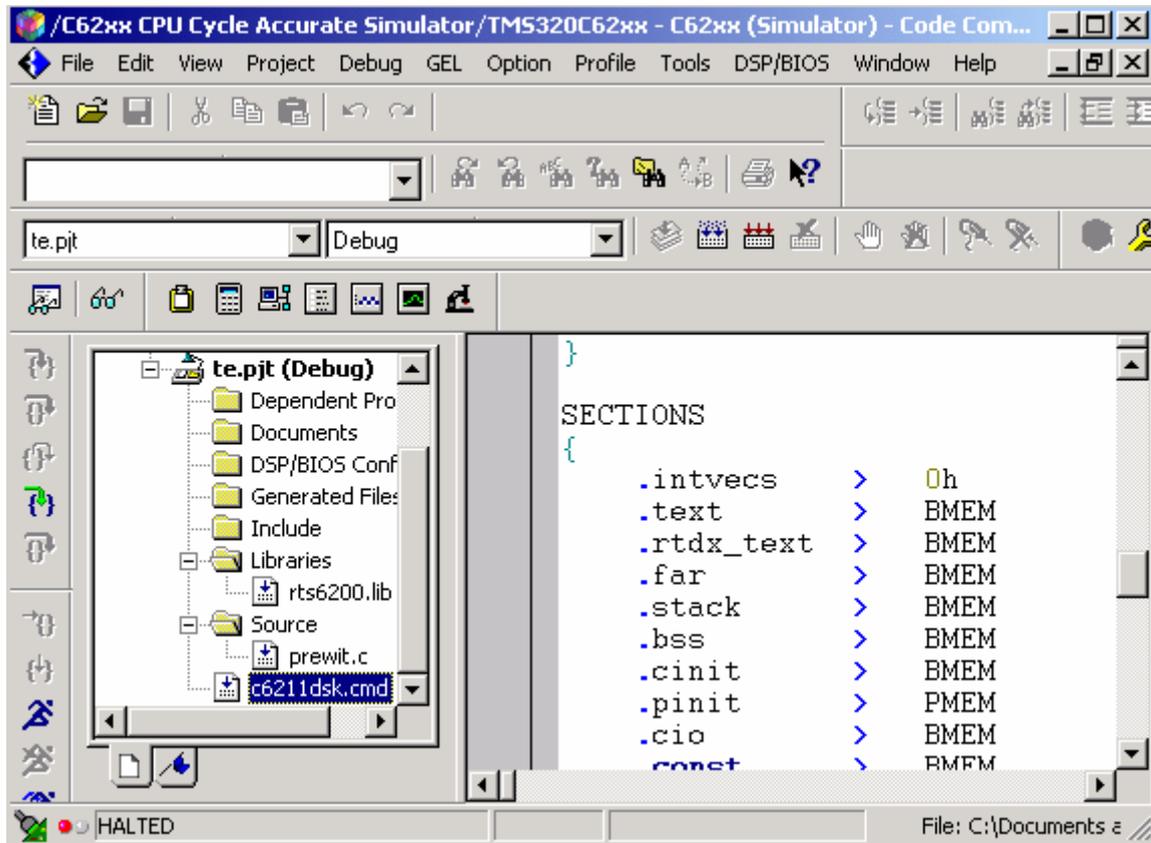


Figure VII.3 Fenêtre principale du CCS

VII.3.2 AJOUT DE FICHIERS AU PROJET

On peut ajouter plusieurs et différents fichiers à notre projet. La procédure à suivre pour ajouter des fichiers à notre projet est la suivante:

- **Etape 1** : on sélectionne Project → Add Files to Project, ou on clique par le bouton droit sur le nom du projet affiché sur la fenêtre puis on sélectionne « Add Files». La boîte de dialogue suivante apparaît :

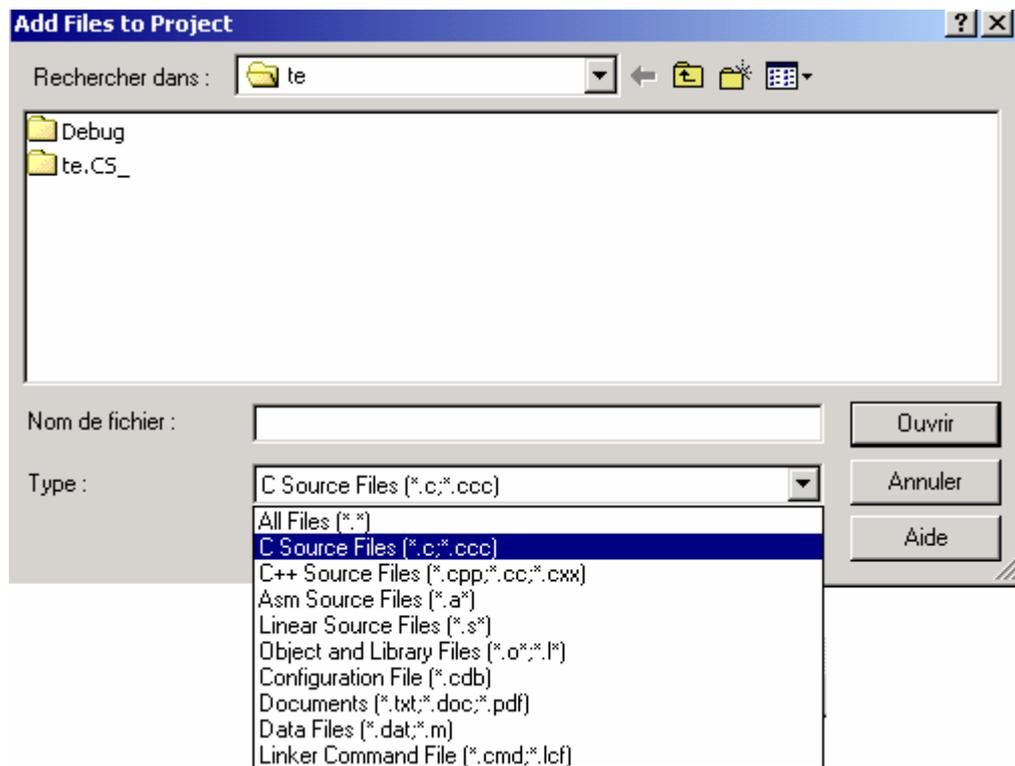


Figure VII.4 Boite de dialogue « add files to project »

- **Etape 2** : dans cette boîte de dialogue on spécifie le fichier qu'on désire ajouter. Si le fichier n'existe pas dans le répertoire courant on parcourt jusqu'à trouver son chemin.

Note : Ne jamais essayer d'ajouter manuellement les fichiers en-têtes/Include (*.h) au projet, ces fichiers seront ajoutés automatiquement lorsqu'on clique sur «Scan All Dependencies »

- **Etape 3** : on clique sur « Open » pour ajouter le fichier spécifié au projet.

L'affichage du projet est automatiquement mis à jour lorsqu'un fichier est ajouté.

Le manager du projet organise les fichiers dans les dossiers : Source, Include, Librairies, et DSP/BIOS Configuration

VII.3.3 EXECUTION D'UN PROGRAMME

Pour exécuter un programme on suit les étapes suivantes:

- **Etape 1** : On sélectionne Project → Rebuild All ou on clique dans la barre d'outils sur le bouton .

Le CCS IDE recompile, réassemble, et relie tous les fichiers dans notre projet. Des messages sur la progression du processus d'exécution sont affichés dans un cadre au fond de la fenêtre.

- **Etape 2** : Par défaut, le fichier exécutable « .out » est construit dans le dossier « Debug » du projet courant, pour changer cette location on sélectionne une location différente dans la barre d'outils du CCS.

- **Etape 3** : On sélectionne File→ Load program, ensuite on sélectionne le programme que nous venons de créer (.out), et on clique sur « Open ». Le CCS IDE charge le programme sur la cible DSP et ouvre une fenêtre « Dis-Assembly » qui montre les instructions traduites en assembleur composant le programme.
- **Etape 4**: On choisit View/Mixed Source/ASM, cela permet de voir notre code source écrit en C et sa traduction en assembleur simultanément.
- **Etape 5** : On sélectionne Debug→ Go to Main pour commencer l'exécution à partir de Main (ie programme principal). L'exécution s'arrête au Main et est identifiée par une flèche.
- **Etape 6** : On sélectionne Debug→ Run ou on clique sur le bouton (bouton  sur la barre d'outils).
- **Etape 7** : On sélectionne Debug→ Halt pour arrêter l'exécution du programme

VII.3.4 CHARGEMENT ET SAUVEGARDE DES DONNEES

Le CCS nous permet de charger et de stocker des données sous format « *.dat » par trois différents moyens :

En utilisant Load/Save à partir du menu :

On sélectionne File→ Data→ Load pour charger les données manuellement à partir 'un fichier « *.dat » qui doit contenir une ligne d'en-tête du DSP formée de 5 champs :

- Le nombre 1651 pour identifier le *.dat ;
- Le format qui est un entier entre 1 et 4, spécifiant le format des données dans le fichier (1= hexadécimal, 2= entier, 3= entier long, 4= nombre flottant);
- L'adresse de départ du bloc qui a été sauvegardé ;
- Le numéro de page correspondant à la page source du bloc ;
- La longueur du bloc.

Nous avons utilisé l'en-tête 1651 2 0 0 0 0 pour toute nos données.

On doit préciser la variable de destination (ou son adresse mémoire) de notre chargement et le nombre d'échantillons à charger.

Pour la sauvegarde des données traitées, on sélectionne File→ Data→ Save, on précise la variable qu'on veut sauvegarder (ou son adresse mémoire) et le nombre d'échantillons à sauvegarder.

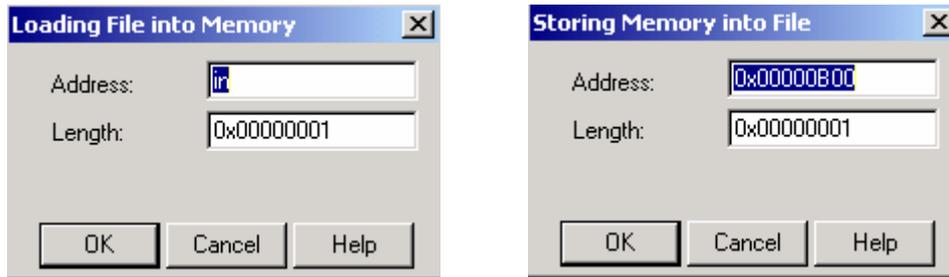


Figure VII.5 Chargement et sauvegarde de données

- **En utilisant les probe-points (les sondes) :**

Ces sondes servent aussi à charger et sauvegarder manuellement les fichiers de données input/output (*.dat), elles seront placées sur le curseur positionné sur le code source de notre projet, il suffit de cliquer sur le bouton . Ensuite on clique sur File → File I/O on obtient la fenêtre (figure VII.6) sur laquelle on choisit les fichiers à connecter aux points sondes pour le chargement (Input) ou pour la sauvegarde (output).

Ces fichiers doivent contenir l'en-tête expliquée précédemment.

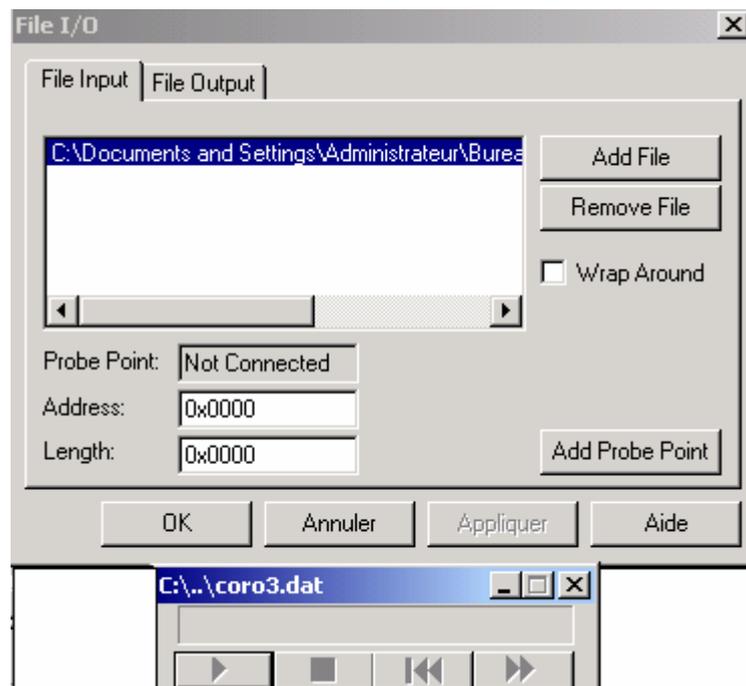


Figure VII.6 Boite de dialogue I/O

Il suffit de cliquer sur  pour charger ou sauvegarder les données.

Remarque : si notre fichier source contient des données qui se répètent dans le temps, on sélectionne l'option « wrap around » sur la boîte de dialogue, par exemple cas d'un signal sinusoïdal.

- **En utilisant les instructions de la directive « `stdio.h` » dans le code source:**

Les instructions : `fopen`, `fscanf`, `fprintf`, `fclose`,...etc, servent à charger et sauvegarder automatiquement les fichiers de données input/output (*.dat). Ces fichiers lus ne doivent pas contenir l'en-tête du DSP.

VII.4 SIMULATION

VII.4.1 FILTRAGE

Nous appliquons différents filtres à une image angiographique . Les résultats sont représentés par la figure VII.7

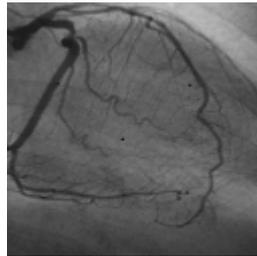
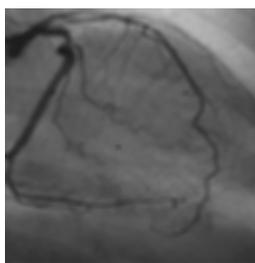


Image originale



Filtre Moyen



Filtre Gaussien



Filtre Médian

Figure VII.7 Résultats pour différents filtres

VII.4.2 DETECTEURS DE CONTOURS

Maintenant on va essayer de simuler les différents détecteurs de contours déjà étudiés, la figure VII.8 montre les résultats obtenus.

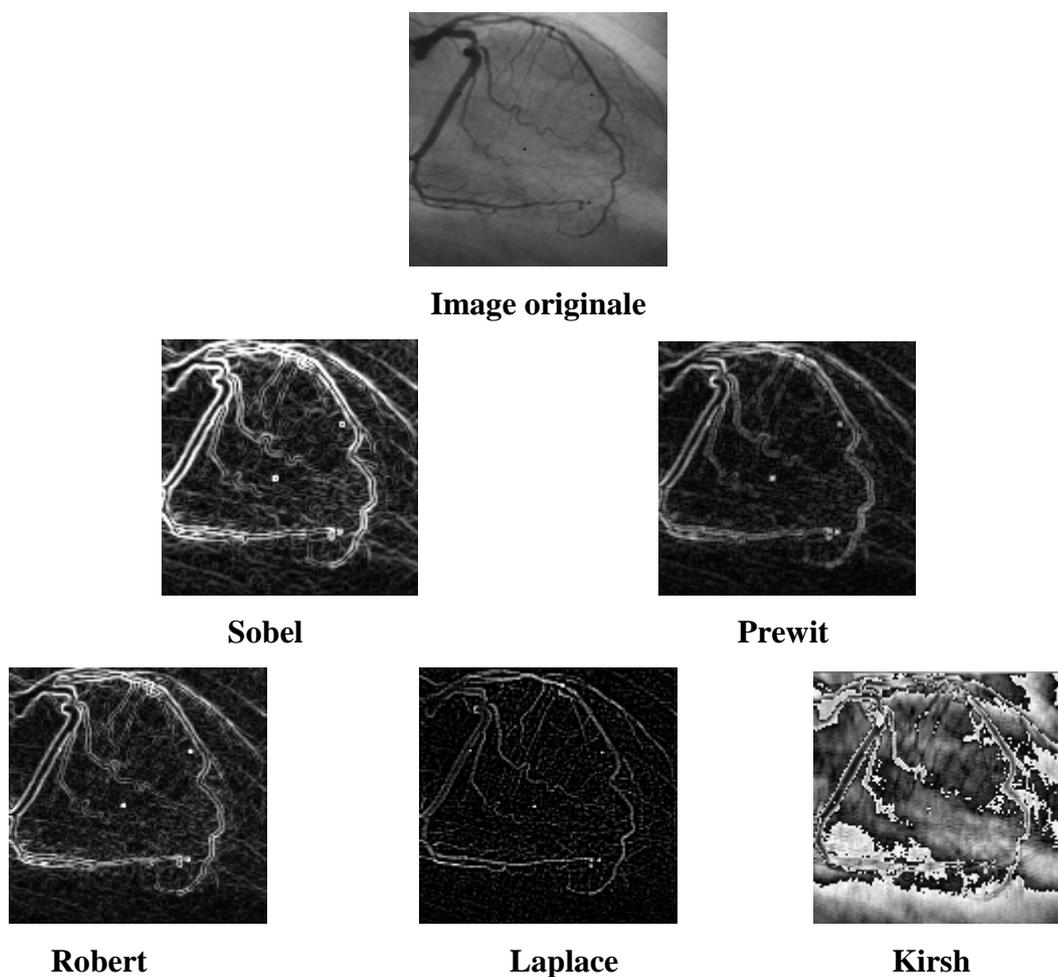


Figure VII.8 Résultats des différents détecteurs de contours

VII.4.3 BINARISATION

Nous représentons ici le résultat de l'algorithme de binarisation obtenu avec le CCS. Nous précisons que ce résultat n'a pas été très convaincant, surtout pour être utilisé pour une squelettisation, d'où le choix d'une autre méthode qu'on a déjà expliqué au chapitre de squelettisation, et qui a été réalisé non pas par CCS mais par le Builder C++..

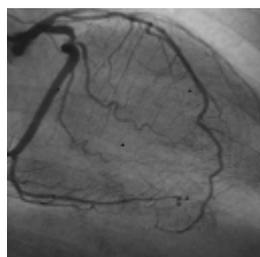


Image originale



**Image Binarisée
(Seuil automatique)**



**Image Binarisée
Seuil= 75**



**Image Binarisée
Seuil= 72**

Figure VII.9 : Binarisation

VII.4.4 SQUELLETISATION

Pour la squelettisation on a choisi d'implémenter deux algorithmes : Zhang & Suen, et Marthon. La figure VII.10 représente les résultats obtenus :

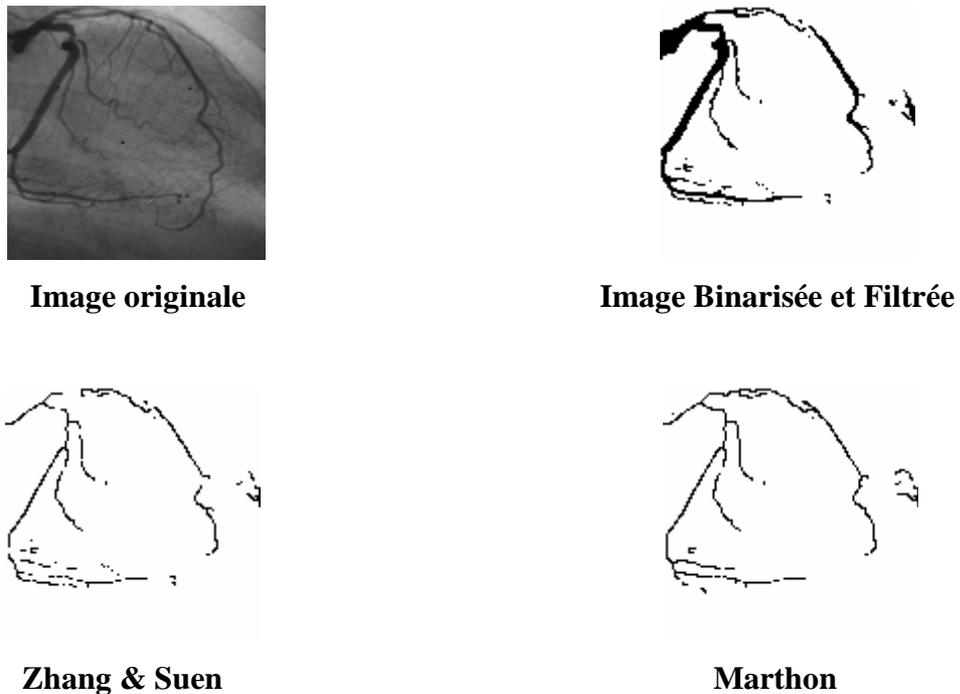


Figure VII.10 Résultats de la squelettisation

Il est bien évident que l'algorithme de Marthon donne des résultats plus convaincants que Zhang et Suen, et comme nous l'avons déjà expliqué au chapitre de squelettisation, c'est cet algorithme qu'on a choisi pour aborder l'étiquetage.

V.4.5 L'ETIQUETAGE

Comme nous l'avons déjà exposé au chapitre d'étiquetage, notre programme comporte trois parties : détection des points de bifurcation, suivi du squelette, et étiquetage. Nous allons détailler le programme (exécuté sous CCS).

```

#include <stdio.h>
#include <stdlib.h>
int in[16384],out[16384]; // Définition des vecteurs d'entrée et sortie
short i,k,j,m,n; // variables globales

```

Cette partie concerne la déclaration des bibliothèques utilisée, ainsi que les vecteurs d'entrée et de sortie (de longueur 128×128) et des variables globales que le programme utilisera.

```

void bif ()
{
    for (i=0;i<128;i++)
    for (j=0;j<128;j++)
    {
        if ((in[128 * (i)+j+1]==0 )&&(in[128 * (i+1)+j]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i)+j-1]==0 )&&(in[128 * (i+1)+j]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i+1)+j-1]==0 )&&(in[128 * (i+1)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i+1)+j+1]==0 )&&(in[128 * (i-1)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i)+j+1]==0 )&&(in[128 * (i+1)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i-1)+j]==0 )&&(in[128 * (i)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i-1)+j]==0 )&&(in[128 * (i+1)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
        if ((in[128 * (i-1)+j+1]==0 )&&(in[128 * (i+1)+j+1]==0)&&(in[128 * (i)+j]==0) )
            out[128 * (i)+j]=255 ;
    }
}

```

Ensuite nous avons défini une fonction **bif** qui a pour rôle « d'éteindre » les point de bifurcation. En effet en faisant un balayage de l'image, on teste les conditions pour rencontrer un point de bifurcation, et quand c'est le cas ce point est mis à la valeur 255 (blanc).

```

void nivgris (n,m)    // fait le suivi de l'artere,et change son niveau de gris
{
    v1 : if (out[128 * (n+1)+m-1]==0)
        {
            out[128 * (n+1)+m-1]=100 * k;
            n++;m--;
            goto v1;
        }

    v2 : if (out[128 * (n)+m+1]==0)
        { out[128 * (n)+m+1]=70 * k;
          m++;
          goto v2; }
    v3 : if (out[126 * (n+1)+m+1]==0)
        { out[128 * (n+1)+m+1]=70 * k;
          m++; n++;
          goto v3; }
    v4 : if (out[128 * (n+1)+m]==0)
        { out[128 * (n+1)+m]=70 * k;
          n++;
          goto v4; }
    v5 : if (out[128 * (n)+m-1]==0)
        { out[128 * (n)+m-1]=70 * k;
          m--;
          goto v5; }

    k=k+10;
}

```

La première fonction de **nivgris** , quand elle est sollicitée , est de chercher le point allumé suivant, en utilisant des boucles, et en testant les points voisins, elle arrive à faire un suivi de l'artère, jusqu'à rencontrer un point blanc qui ne peut être que le bout de l'artère ou un point de bifurcation (déjà effacé par la fonction bif)

La deuxième fonction de **nivgris** consiste en l'attribution d'un niveau de gris spécifique pour chaque branche suivie. En effet elle met le vecteur sorti out à une valeur 70 * k avec k incrémenté pour chaque passage à une nouvelle branche. Ces différences de niveaux de gris sont notre moyen d'étiqueter le squelette.

```

void main ()
{
    k=1 ;
    for (i=0;i<128;i++)
    for (j=0;j<128;j++)

    out[128 * i+j]= in[128 * i+j];

    bif (); // effacer les points de bifurcation

    for (i=0;i<128;i++)
    for (j=0;j<128;j++)
    if (out[128 * (i)+j] > in[128 * (i)+j])
    {
        n=i;m=j;

        nivgris (n,m); // suivi et étiquetage
    }
}

```

Le programme principal fait un balayage de l'image afin de mettre le vecteur de sortie aux mêmes valeurs que celui de l'entrée.

Ensuite il fait appel à `bif`, cette dernière se chargera de mettre les points de bifurcation à 255 (blanc).

Il refait un autre balayage de l'image, et s'il y a une différence entre les vecteur entrée et le vecteur sortie, c'est qu'il a rencontré un point de bifurcation, là il fait appel à la fonction `nivgris`, qui se charge de faire changer le niveau de gris de la branche concernée par le code de Freeman en faisant un suivi de cette branche, et en changeant pour chaque pixel son niveau de gris.



Squelette utilisé



Squelette étiqueté

Figure VII.11 Résultat de l'étiquetage par niveaux de gris

VII.5 CONCLUSION

Ce chapitre nous a permis de voir comment simuler les différents algorithmes développés dans notre travail. Les résultats sont liés. En effet les algorithmes travaillaient en chaîne, et chaque résultat était utilisé par l'algorithme suivant.

Nous avons essayé d'obtenir le meilleur squelette possible, car sa qualité est déterminante pour l'étiquetage. Ce dernier a été fait par des niveaux de gris, en utilisant le CCS (programmé en C). Nous avons obtenu une image étiquetée mais l'étiquetage ne s'est pas étendu vers toute l'artère.

CONCLUSION GENERALE

Grâce à notre travail nous avons pu nous initier à un domaine très important du traitement du signal en général et des images d'angiographie par rayons X en particulier, notre cible était le réseau des artères coronaires, et notre but était la contribution à la conception d'un système d'aide embarqué pour la détection des sténoses ou thromboses.

Après avoir introduit la problématique médicale, nous avons essayé d'aller étape par étape jusqu'à l'étiquetage de l'arbre coronaire. Nous avons commencé par introduire des notions générales liées au traitement des images, en passant par les prétraitements comme le filtrage et l'amélioration de contraste, puis un état de l'art sur les différents détecteurs de contours. L'obtention du squelette même s'il n'est pas parfait était la phase la plus délicate, où on a contourné le problème lié l'extraction de l'arbre par une détection de contour suivie d'un traitement morphologique ou dilatation puis amincissement de l'arbre obtenu, cette procédure a donnée des résultats nettement meilleurs par rapport à une binarisation par seuillage.

Une fois l'image binarisée, nous avons appliqué deux algorithmes de squelettisation, un à critère topologique (Zhang et Suen), l'autre à critère métrique (Marthon). Le meilleur résultat fut celui de Marthon, en effet le squelette ne présentait pas trop de discontinuités.

Pour l'étiquetage, nous avons d'abord essayé d'utiliser le Builder C++, et nous avons appliqué le programme à un squelette « type » puis squelette réel. L'étiquetage s'est fait sans problèmes pour les régions où le squelette était bien défini.

Au dernier chapitre nous avons expliqué en détail les différentes étapes pour simuler nos algorithmes sur CCS, l'exemple qu'on a développé était celui de l'étiquetage, que nous avons réalisé par une coloration en niveaux de gris.

PERSPECTIVES

- Utilisation des étiquettes obtenues pour une reconstitution 3D de l'arbre coronaire.
- Utilisation de l'approche multirésolution pour l'extraction du squelette sans passer par les détecteurs de contours

Bibliographie

- [1] Raman, S., Magorien, R., Vaillant, R., Payne, P., Benali, “ Rotational Cardiovascular X-Ray Imaging for Left Coronary Artery Angiography Using a Digital Flat-Panel Cardiac Imaging System.”American Journal of Cardiology, 2002
- [2] Christophe BLONDEL, « Modélisation 3D et 3D+t des artères coronaires à partir de séquences rotationnelles de projections rayons X ». Thèse de doctorat de l’université de Nice-Sophia Antipolis. 2004.
- [3] Amiel, M., Maseri, A., Petitier, H., et Vasile, “ Coronary artery diseases : diagnostic and therapeutic imaging approaches” . Springer-Verlag. 1984
- [4] Alain CHEHIKIAN, Philippe BOLON, « Analyse d’image, Traitements de bas niveau, Technique de l’ingénieur ».
- [5] M.Kunt, “Traitement numérique des images”, PPUR, 1993.
- [6] Jean-pierre Cocquerez et sylvie PHILIP, « Analyse d’image : filtrage et segmentation », Springer Verlag, 1996
- [7] Walter G.Kropatsch, Horst BISCHOF, “Digital Image Analysis“, SpringerVerlag. 2001
- [8] Rachid DERICHE , « Techniques d’extraction de contours » , INRIA Sophia-Antipolis 2004.
- [9] S.TIRES, N. ROULA, ”Implémentation d’algorithmes de traitement d’images sur DSP Application à la séparation des Chromosomes”, PFE Ecole Nationale Polytechnique, 2006.
- [10] http://www710.univ-lyon1.fr/~fdenis/club_EEA/cours
- [11] AHMED ZAID Salim, « Implémentation d’algorithmes de traitement d’images sur DSP C6000 » PFE Ecole Nationale Polytechnique, 2005.
- [12] H.KRAUS, ” Scan & retouche d’images”, CAMPUS Press, 1999 .
- [13] P. Canham Van Dijken, « L’image numérique », Institut de pathologie de Lausanne, 2001.
- [14] A.MESTAR, R.KHELIFI, « Segmentation des images coronarographiques : Application à la détection des cardiopathies », PFE Ecole Nationale Polytechnique,2005.
- [15] H. AIT ABDESSLAM, « Réalisation d’un système d’aide à la détection des aberrations Chromosomiques », Thèse de magister ,Ecole Nationale Polytechnique, 2005.
- [16] H. Konik , B. Laget, M. Calonnier , « Segmentation d’images par utilisation de pyramides à bases locales » 1993

- [17] M. YOUNSI Rafik, « Segmentation des images en angiographie par rayon X en vue d'un étiquetage automatique de l'arbre coronaire », PFE , USTHB, 2006.
- [18] Rulf CHASSAING. "DSP Using C and application and the TMS320C6x DSK" John Wiley & Sons, 2002.
- [19] TMS320C6000 Peripherals, SPRU190d.pdf, Texas Instruments.
- [20] TMS320C6000 Programmer's Guide, SPRU198d.pdf, Texas Instruments.
- [21] TMS320C6000 Optimizing Compiler User's Guide, SPRU187g.pdf, Texas Instruments.
- [22] R. Haralick and L. Shapiro, "Computer and Robot Vision", Addison-Wesley Publishing Company, 1992..
- [23] Djamel MERAD, « Reconnaissance 2D/2D et 2D/3D d'objets à partir de leurs squelettes ». Thèse de doctorat de L'Université d'Evry-Val d'Essonne.2004
- [24] CHIH-YANG LIN, YU-TAI CHING. « Extraction of coronary arterial tree using cine X-Ray angiograms", National Chiao Tung University, Hsin Chu, Taiwan. 2005
- [25] Yaowu Xu ."Object Segmentation and Image Labeling by Learning from Examples".University of Rochester Rochester, New York , 2003
- [26] E. Davies, "Machine Vision: Theory, Algorithms and Practicalities", 1990..

ANNEXES

ANNEXE I

AUTRES MODALITES D'ACQUISITION D'IMAGES CARDIAQUES

En l'état actuel, l'angiographie par rayons X reste la modalité de choix pour l'exploration des pathologies coronaires par les médecins. Néanmoins, de nombreuses possibilités sont offertes par d'autres modalités d'imagerie cardiaque que nous décrivons maintenant brièvement.

I.1 Imagerie cardiaque anatomique

I.1 .1 Scanner CT

Le scanner CT (Computed Tomography) produit des modèles par coupe de densité des tissus. L'introduction plus récente des acquisitions spiralées et des détecteurs multi-barettes permet de reconstruire directement des images volumiques. Les différences de densité permettent de discriminer les différents organes observés et de déterminer leur anatomie. En dépit d'avancées récentes importantes, ses difficultés dans l'imagerie des artères coronaires résident dans les résolutions spatiale et temporelle encore trop faibles (typiquement 0,75 millimètre et 100 millisecondes par coupe) pour pouvoir observer clairement les artères coronaires, qui sont des structures fines, animées d'un mouvement rapide. Toutefois, un avantage du scanner CT sur l'angiographie par rayons X est sa capacité à mieux discriminer les dépôts calcifiés (durs) des dépôts lipidiques (mous). En effet, les longueurs d'onde des rayons X mis en jeu pour le scanner CT diffèrent de ceux utilisés en angiographie par rayons X et sont plus adaptés à cette discrimination. La figure 1.11 montre un exemple de visualisation d'une reconstruction des artères coronaires par un scanner CT, au meilleur de l'état de l'art actuel.

I .1.2 Scanner EBCT

Le scanner à faisceau d'électrons ou EBCT (Electron Beam Computed Tomography) utilise une technologie différente de celle du scanner pour la génération des rayons X. Celle-ci permet une résolution temporelle très fine, mais obtenue au détriment d'une résolution spatiale qui est trop grossière (en particulier dans la direction axiale où elle est typiquement de 3 millimètres) pour permettre de bien caractériser l'anatomie des coronaires. En outre, cet équipement est bien plus cher qu'un angiographe ou qu'un scanner CT. Le scanner EBCT permet en revanche une excellente évaluation du taux de calcification dans les dépôts artériels, c'est d'ailleurs l'utilisation principale qui en est faite en routine clinique. La figure 1.12 montre un exemple de visualisation d'une reconstruction des artères coronaires par un scanner EBCT.

I.1.3 Angiographie par résonance magnétique

L'imagerie de type ARM (Angiographie par Résonance Magnétique) ou MRA (Magnetic Resonance Angiography) utilise les propriétés magnétiques des protons pour caractériser la

présence de molécules d'eau. On peut donc identifier la répartition et le mouvement de ces molécules dans le milieu observé. En utilisant l'écoulement sanguin, des techniques dites de temps de vol et de contraste de phase permettent d'observer les artères coronaires. De même que pour le scanner CT, l'acquisition de type ARM n'offre pas encore une résolution spatiale suffisamment fine pour permettre d'explorer précisément l'anatomie des artères coronaires. Par contre, cette modalité est bien adaptée à l'étude de l'anatomie du myocarde et de sa dynamique. La figure 1.13 montre un exemple de visualisation des artères coronaires par imagerie ARM avec injection d'un produit de contraste.

I.1.4 Imagerie ultrasonore intravasculaire

L'examen des artères coronaires par ondes ultra-sonores ne peut pas se faire depuis l'extérieur du corps car les images obtenues ne permettent absolument pas de visualiser les artères coronaires. C'est pour cela que la sonde à ultrasons est amenée par voie interne, le plus souvent par voie fémorale. L'imagerie ultrasonore intravasculaire ou IVUS (Intra Vascular Ultra Sound) donne des images du gradient spatial d'impédance acoustique dans l'ensemble des directions perpendiculaires à l'axe de la sonde. Typiquement, cela permet d'observer les interfaces entre tissus (où l'impédance acoustique change) et donc de délimiter les différentes structures présentes. Dans le cas des artères coronaires, cela permet de discriminer les différentes couches de la paroi du vaisseau : lumen, intima, media et adventitia, ainsi que les dépôts graisseux ou calcifiés. La figure 1.14 montre un exemple de visualisation des artères coronaires par imagerie de type IVUS.

Les images obtenues par IVUS ne permettent pas de donner l'anatomie des artères coronaires car la sonde suit la géométrie des artères et ne donne pas d'information de position. L'information obtenue est locale et ne permet pas de représentation géométrique des artères coronaires en dehors de la coupe elle-même. On peut néanmoins pallier cette limitation en combinant l'imagerie IVUS avec l'imagerie radiologique pour localiser la sonde ultrasonore et construire un modèle issu des images ultrasonores recalé dans l'espace selon la géométrie donnée par l'angiographie biplan.

I.1.5 Tomographie par cohérence optique

La tomographie par cohérence optique ou OCT (Optical Coherence Tomography) est une autre modalité d'imagerie intravasculaire dont l'introduction est récente. Elle utilise l'interaction entre lumière et matière pour produire des images caractéristiques de la transparence et de la structure des matériaux. Les images produites s'apparentent à celles données par l'imagerie ultrasonore. Cette nouvelle modalité, qui reste actuellement en phase d'études, offre une meilleure résolution spatiale que l'imagerie ultrasonore et semble en particulier donner de meilleurs résultats en terme de sensibilité pour l'évaluation post-opératoire de la pose de stents.



Fig. 1.1 : Vue projective 2D d'une reconstruction 3D des artères coronaires par un scanner CT.

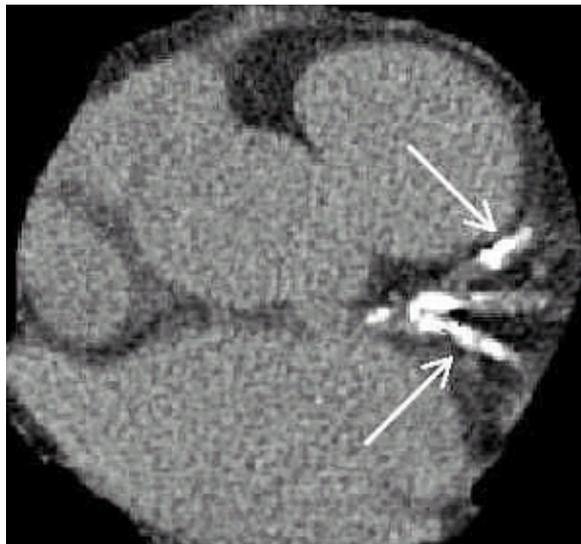


Fig. 1.2 : Coupe d'une reconstruction des artères coronaires par un scanner EBCT. Des dépôts calcifiés sont indiqués en blanc.



Fig. 1.3 : Coupe d'une reconstruction des artères coronaires par imagerie ARM.

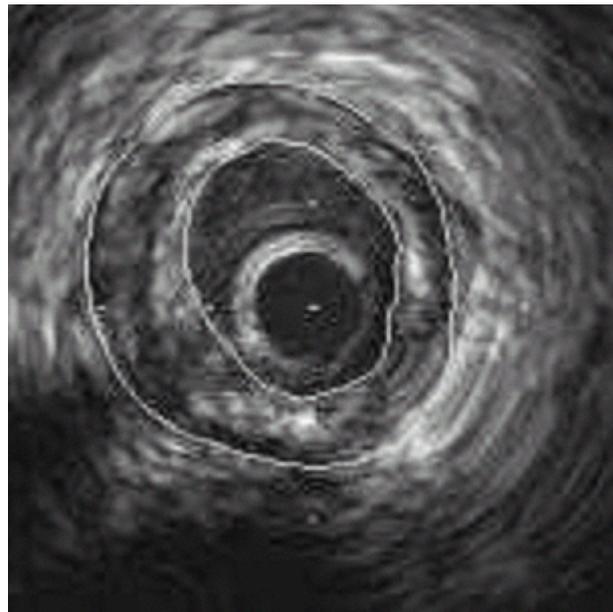


Fig. 1.4 : Visualisation des artères coronaires par imagerie IVUS. L'intima et l'adventitia sont délimitées en blanc. On peut voir, dans ce cas pathologique, l'épaississement de la paro entraîné par l'athérosclérose.

I.2 Imagerie cardiaque fonctionnelle

De manière générale, l'imagerie fonctionnelle ne procure pas des résolutions spatiale et temporelle suffisantes pour l'observation des artères coronaires. Cependant, la synchronisation de l'acquisition avec le signal ECG permet d'accéder à des résolutions temporelles suffisamment fines pour l'imagerie cardiaque. Pour cela, l'ouverture du détecteur est asservie à la détection d'une phase cardiaque spécifique, qui est identifiée à partir du signal ECG. Toutes les données récoltées le sont donc à la même phase et présentent donc une

meilleure cohérence temporelle. Nous présentons brièvement deux modalités intéressantes car elles caractérisent les effets des pathologies coronaires et permettent d'identifier la nérose du tissu myocardique. Les modalités fonctionnelles nous renseignent non pas sur le type d'un tissu, mais sur sa qualité, en particulier du point de vue métabolique. Un tissu mal irrigué aura une activité métabolique (respiration cellulaire) moindre que celle du tissu sain correspondant. Ce fait simple permet de localiser et de quantifier l'impact des lésions coronariennes sur la fonction cardiaque.

Les modalités les plus courantes sont le TEMP (Tomographie d'Emission Mono Photonique) ou SPECT (Single Photon Emission Computed Tomography) et le TEP (Tomographie d'Emission de Positons) ou PET (Positron Emission Tomography). Un traceur radioactif est intégré dans une molécule physiologique, utilisée comme vecteur et choisie pour être naturellement métabolisée par le tissu à examiner. Après injection du traceur dans la circulation sanguine, la molécule est fixée par l'organe cible, sa concentration locale dépendant de l'activité métabolique du tissu et de son irrigation sanguine par le système circulatoire. Les photons issus de l'activité radioactive de l'isotope sont détectés sur un capteur plan. A partir de ces données projectives, une reconstruction de la localisation des désintégrations radioactives est réalisée. Cette intensité d'émissions radioactives est ensuite corrélée à la concentration de la molécule physiologique, puis à l'activité métabolique des tissus.

La connaissance des territoires d'irrigation des artères coronaires permet ensuite de lier territoire hypoperfusé, ayant un métabolisme anormalement bas, et artère lésée. La figure 1.15 montre un exemple de visualisation d'un ventricule gauche pathologique par imagerie fonctionnelle de type SPECT.

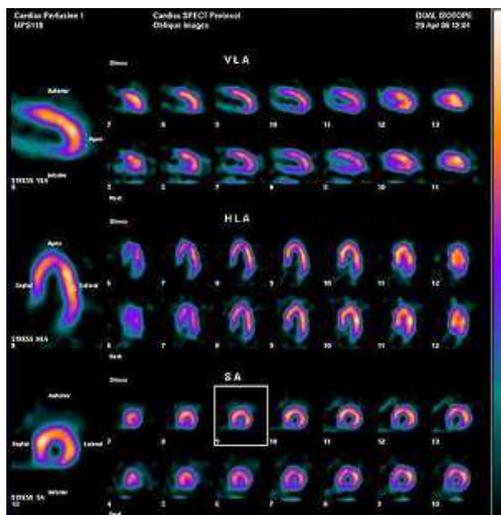


Fig. 1.15 : Visualisation de coupes d'un ventricule gauche pathologique reconstruit par imagerie fonctionnelle SPECT.

ANNEXES II

DETERMINATION DU MASQUE GRADIENT

Soit $h(x,y)$ la réponse impulsionnelle de l'opérateur appliqué sur la fenêtre $(2M+1)*(2N+1)$ de l'image. $h(x,y)$ est donc un tableau de valeurs de dimensions $(2M+1)*(2N+1)$ appelé « masque ».

Appliquer un masque en un point (x,y) de l'image revient à remplacer l'éclairement de ce point $I(x,y)$ par la convolution de $h(x,y)$ avec l'image s'écrit :

$$I(x,y) = \sum_{m=-M}^M \sum_{n=-N}^N h(m,n) I(x-m, y-n) \quad (1)$$

Considérons maintenant l'exemple du détecteur des transitions verticales grad y. la transformée de Fourier de cet opérateur appliqué à l'image est égale à :

$$\text{TF}\left[\frac{\partial}{\partial Y} I(x,y)\right] = j u_y \text{TF}[I(x,y)] \text{ (propriétés de la TF) .}$$

De cette relation et de la relation espace-fréquence , on déduit que :

$$H(e^{jux}, e^{juy}) = j u_y. \quad (2)$$

Cette transformée est indépendante de la variable u_x . Par conséquent le développement de cette transformée en série de Fourier bidimensionnelle est équivalent a une série de Fourier unidimensionnelle par rapport à la variable u_y . Cette série de Fourier est donnée par :

$$j u_y = \sum_{n=-N}^N C_n e^{jnuy} \quad \text{avec} \quad C_n = \frac{n}{2\pi} \int_{-\frac{\pi}{n}}^{\frac{\pi}{n}} j u_y e^{jnuy} d u_y. \quad (3)$$

Le calcul de C_n donne :

$$\begin{aligned} C_n &= 1/n && \text{pour } n \neq 0 \\ C_n &= 0 && \text{pour } n = 0 \end{aligned} \quad (4)$$

Comme cette transformée est indépendante de la variable u_x , on peut alors écrire, dans le domaine bidimensionnel la série de Fourier sous cette forme :

$$H(e^{jux}, e^{juy}) = \sum_{m=-M}^M \left[\sum_{n=-N}^N (1/n) e^{jnuy} + 0 \right] e^{jmux} \quad \text{avec } n \neq 0. \quad (5)$$

Sachant que la transformée de Fourier discrète $H(e^{jux}, e^{juy})$ de $h(m,n)$ est donnée par :

$$H(e^{jux}, e^{juy}) = \sum_{m=-M}^M \sum_{n=-N}^N h(m,n) e^{jmux} e^{jnuy} \quad (6)$$

Les valeurs de $h(m,n)$ sont alors déduites par identification.

Pour $n \neq 0$ $h(m,n) = 1/n$ et pour $n = 0$ $h(m,n) = 0$

Lorsque $N=1$ et $M=1$, c'est-à-dire pour une fenêtre 3×3 , $h(m,n)$ prend les valeurs suivantes :

$$\begin{array}{lll} h(-1,-1) = -1 & h(-1,0) = 0 & h(-1,1) = 1 \\ h(0,-1) = -1 & h(0,0) = 0 & h(0,1) = 1 \\ h(1,-1) = -1 & h(1,0) = 0 & h(1,1) = 1 \end{array}$$

le masque s'écrit alors :

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

On retrouve un masque classique permettant de détecter les contours verticaux.

Cette procédure nous donne un moyen rigoureux pour choisir les coefficients du masque. elle peut être généralisée à tous les détecteurs basés sur le gradient et pour toutes les directions possibles.

Cependant les coefficients de ces masques peuvent être ajustés pratiquement, selon besoins, c'est d'ailleurs de façon totalement empirique que la plupart des masques courants ont été introduits.

ANNEXES III

MODES D'ADRESSAGE ET JEU D'INSTRUCTION DU C6000

III.1 LES MODES D'ADRESSAGE LINEAIRES ET CIRCULAIRES

Les modes d'adressages déterminent la façon d'accéder à la mémoire. Ils spécifient aussi la manière avec laquelle on manipule les données, comme la recherche d'un opérande indirectement à partir d'une position mémoire. Sur le C6x les deux modes d'adressage, linéaire et circulaire, sont supportés. Le plus communément utilisé est le mode d'adressage indirect de la mémoire.

III.1.1 Adressage indirect

L'adressage indirect peut être utilisé avec ou sans déplacement. Le registre R représente un des 32 registres de A et B, qui peut se pointer sur les adresses mémoires. A proprement parler, ces registres sont des pointeurs. Le mode d'adressage indirect utilise " * " en conjonction avec l'un des 32 registres. Pour illustrer, considérons le registre d'adresse R.

1. *R. Le registre R contient l'adresse de la position mémoire où la valeur de la donnée est stockée.
2. *R++(d). Le registre R contient une adresse. Après son utilisation, le registre R est post-incrémenté telle que la nouvelle adresse est l'adresse actuelle avec un déplacement d. Le même raisonnement est valable dans le cas (--).
3. *++R(d). L'adresse est préincrémentée d'un déplacement d, telle que l'adresse courante est R+d. Un double ' - ' introduit une prédécrémentation de l'adresse mémoire telle que la courante sera R-d.
4. *+R(d). L'adresse est préincrémentée par d telle que l'adresse courante est R+d. Cependant, dans ce cas, R est préincrémenté sans modification. Contrairement au cas précédent, R n'est pas mis à jour ou modifié.

III.1.2 Adressage circulaire

L'adressage circulaire est utilisé pour créer un buffer circulaire au niveau matériel, qui est fréquemment utilisé par plusieurs algorithmes du traitement du signal. Exemples, les filtres numériques et corrélations.

Le C6x a une architecture dédiée mettant en oeuvre l'adressage circulaire, qui peut être utilisé en conjonction avec un buffer circulaire afin de mettre à jour les échantillons en remplaçant l'ancien échantillon par le nouveau et éviter le décalage direct de données. Lorsque le pointeur atteint la position fin du buffer circulaire qui contient le dernier élément (échantillon) du buffer, et après son incrémentation, le pointeur automatiquement pointe sur la position début du buffer circulaire qui contient le premier élément.

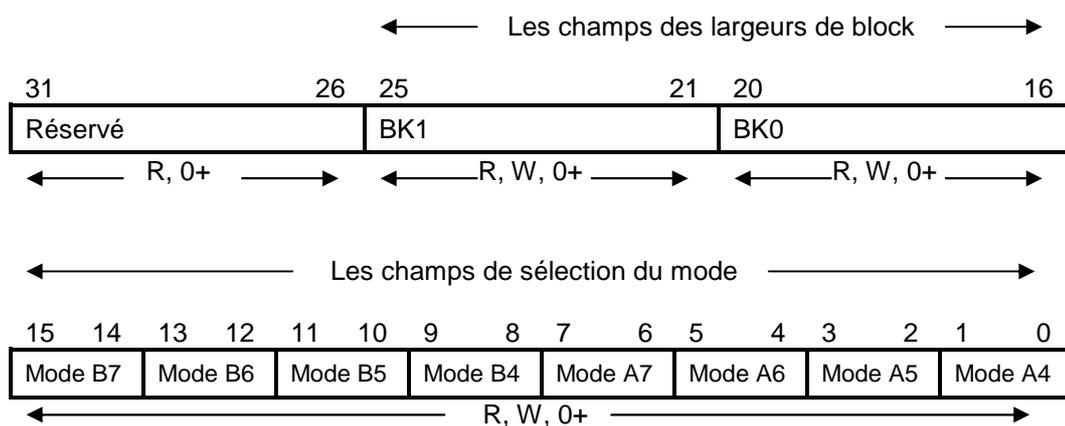


FIGURE III.1 Registre des modes d'adressage

Deux buffers circulaires indépendants sont disponibles en utilisant BK0 et BK1 avec le registre mode d'adressage (AMR). Les huit registres de A4 à A7 et de B4 à B7, en conjonction avec les unités .D, peuvent être utilisés comme pointeurs (NB : tous les registres peuvent être utilisés en adressage linéaire). Le segment de code suivant illustre l'utilisation du buffer circulaire en utilisant le registre B2 pour configurer le AMR (seul la partie B intervient dans la configuration des registres spéciaux) :

```
MVK .S2 0x0004, B2 ;les 16bits faibles de B2 sont affectés. Le A5 est sélectionné
                    comme pointeur
```

```
MVKLH .S2 0x0005, B2 ;les 16bits forts de B2 sont affectés. Sélection de B0, fixer N=5
```

MVC .S2 B2,AMR ;transfert des 32bits de B2 vers le AMR

Les deux instructions MVK et MVKLH, utilisant l'unité .S, font transférer la valeur 0x0004 dans les 16LSBs du registre B2 et la valeur 0x0005 dans les 16MSBs de B2. L'instruction MVC (transfert de constante) est la seule instruction qui peut avoir l'accès au registre AMR et autres registres de contrôle, comme elle est exécutée exclusivement sur la partie B en conjonction avec les registres et unités fonctionnelles de celle-ci. En résumé, une valeur de 32bits créée dans B2, est transférée par la suite au registre AMR via l'instruction MVC.

La valeur 0x0004 = (0100)_B positionne le bit 2 du AMR à 1 (les autres à zéro). Ce qui se traduit par le choix du mode 01 et prendre A5 comme pointeur du buffer circulaire utilisant le block BK0. La table 3.4 présente les modes associés avec les registres pointeurs.

La valeur 0x0005 = (0101)_B positionne le 16ème et 18ème bit du AMR à 1. Cela correspond à la valeur N utilisée pour fixer la taille du buffer i.e $2^{N+1} = 64$ en utilisant le BK0.

Tableau III.2 Description des Modes du AMR

Mode	Description
0 0	Adressage linéaire (par défaut, reset)
0 1	Adressage circulaire utilisant BK0
1 0	Adressage circulaire utilisant BK1
1 1	Réservé

III.2 LE JEU D'INSTRUCTIONS DU TMS320C6x

III.2.1 Format du code assembleur

Un code en assembleur C6x est représenté par le champ suivant :

Label || [] Instruction Unit Operands ;comments

Un label (i.e étiquette), s'il est présent, indique l'adresse ou la position mémoire qui contient l'instruction ou la donnée. L'étiquette doit être sur la première colonne. Les deux barres parallèles (||) sont ajoutées si l'instruction s'exécute en parallèle avec celle qui la précède. Le champ qui suit est optionnel pour rendre l'instruction associée conditionnelle. Pour ce faire, cinq registres sont disponibles à savoir A1, A2, B0, B1 et B2. Par exemple, [A1] indique que l'instruction associée est exécutée si A1 est différent de zéro ; alors qu'avec [!A1] l'instruction associée est exécutée si A1 est égale à zéro. On peut dire que toutes les instructions du C6x sont conditionnelles ; c'est l'une des caractéristiques de l'architecture VLIW. Le champ 'Instruction' peut être soit une directive (i.e une commande pour l'assembleur) ou bien un mnémonique. Exemple :

.word value

Cette directive réserve 32 bits en mémoire pour *'value'*. Le champ *'Unit'*, qui peut être l'une des huit unités fonctionnelles, est optionnel. Ce champ permet de désigner l'unité qui sera responsable de l'exécution de l'instruction. Les commentaires mis sur la première colonne sont soit précédés par un astérisque ou un point virgule, autrement par un point virgule.

III.2.2 Directives 'assembleur'

Une directive 'assembleur' n'est pas une instruction, c'est un message pour l'assembleur (pas le compilateur). Prise en compte lors du processus d'assemblage, la directive n'occupe pas d'espace mémoire comme l'instruction, et ne produit pas un code exécutable.

Les adresses des différentes sections peuvent être spécifiées par des directives. Par exemple,

```
.sect      'my_section'
```

L'assembleur crée diverses sections grâce aux directives, qu'on verra ultérieurement, telles que *.text* pour le code exécutable, *.data* pour les données et *.bss* pour la déclaration des variables.[18]

Autres directives fréquemment utilisées :

.short : initialiser un entier de 16bits.

.int : initialiser un entier de 32bits (aussi *.word* ou *.long*).

.float : initialiser une constante selon le standard virgule flottante IEEE-32bits simple précision.

.double : initialiser une constante selon le standard virgule flottante IEEE-64bits double précision.

