

République algérienne démocratique et populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
المركز الجامعي لعين تموشنت
Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent
Institut de Technologie
Département de Génie Electrique



Projet de fin d'études
Pour l'obtention du diplôme de Master en :
Domaine : SCIENCE ET TECHNOLOGIE
Filière : Electronique
Spécialité : Génie de Télécommunications
Thème

**Hybridation de deux algorithmes pour une détection de la tumeur
cérébrale sur TMS320C6416 DSK**

Présenté Par :

- 1) BOUMEDINE Aammar.
- 2) BENKHEDDA Abdellatif.

Devant le jury composés de :

BEMMOUSSAT	Chemseddine	MCB	C.U.B.B	(Ain Temouchent)	Président
BOUTKHIL	Malika	MAA	C.U.B.B	(Ain Temouchent)	Encadrant
BEDDAD	Boucif	Doctorant	U.D.L.S.B.A	(sidi bel abbes)	Co-encadrant
BENOSMAN	Mourad	MCB	C.U.B.B	(Ain Temouchent)	Examinateur

Année universitaire 2016/2017

Je dédie ce mémoire

Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour toi. Tu m'as comblé avec ta tendresse et affection tout au long de mon parcours .

Tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études, tu as toujours été présente à mes cotés pour me consoler quand il fallait. En ce jour mémorable, pour moi ainsi que pour toi, reçoit ce travail en signe de ma vive reconnaissance et ma profonde estime. Malheureusement tu n'es plus la maman, qu'Allah t'accueille dans son vaste paradis

A mon père, En témoignage de ma gratitude, si grande qu'elle puisse être, pour tous les sacrifices qu'il a consenti pour mon bien être et le soutien qu'il m'a prodigué tout le long de mon éducation, que dieu le tout puissant, le préserve et le procure santé et longue vie.

À ma femme, mon grand amour avec tous mes vœux de bonheur et de prospérité.

À Mon frère Zakaria et a mes sœurs Hafsa Fatima et Assia et ma nièce Salsabila , Pour leur encouragement et leur bonté qu'ils m'ont accordé, j'exprime ma profonde reconnaissance et mon grand respect.

À toute ma famille, mes amis et tous ceux que j'aime.

À mes très chers professeurs qui sont la source de mon inspiration qu'ils trouvent dans ce travail les sincères témoignages de ma profonde affectation et de ma haute considération

ABDELLATIF

Dédicace

Je dédie ce modeste travail :

AU mémoire de mon père

A ma très chère mère

À ma femme qui a toujours été à mes côtés

À tous mes frères et soeurs, ainsi que leurs enfants

A toute ma grande famille

A mon binôme sa participation et son aide.

A mes enseignants et surtout monsieur beddad boucif qui m'a aidé

pendant la durés de réalisé mon travaille.

À tous les étudiants de la promotion génie télécommunication

À tous mes amis et collègues de l'Université et d'ailleurs

AMMAR

Remerciement

Toute notre parfaite gratitude et remerciement à Allah le plus puissant qui nous a donné la force, le courage et la volonté pour élaborer ce Travail. C'est avec une profonde reconnaissance et considération particulière que je remercie mon encadreur pour leur soutien, leurs conseils judicieux et leurs grandes bienveillances durant l'élaboration de ce projet. J'exprime ma reconnaissance à tous les membres de jury d'avoir accepté de lire ce manuscrit et d'apporter les critiques nécessaires à la mise en forme de cet ouvrage. Enfin, à tous ceux qui m'ont aidé de près ou de loin pour la réalisation de ce projet de fin d'étude

TABLE DES MATIERES

INTRODUCTION GENERAL	1
I.1 INTRODUCTION	4
I.2 IMAGE NUMERIQUE.....	4
I.2.1 ACQUISITION, ECHANTILLONNAGE SPATIAL ET QUANTIFICATION:	4
I.2.2CONSTRUCTION D'UNE IMAGE NUMERIQUE:.....	6
I.3 PIXEL.....	7
I.4 CARACTERISTIQUES DE L'IMAGE NUMERIQUE	8
I.4.1 TAILLE D'UNE IMAGE.....	8
I.4.2. RESOLUTION	8
I.4.3 HISTOGRAMME	8
I.4.4 LUMINANCE.....	9
I.4.5 NIVEAU DE GRIS	9
I.5 LES DIFFERENTS TYPES D'IMAGE NUMERIQUE	11
I.6 CODAGE D'UNE IMAGE NUMERIQUE	11
I.7 TRAITEMENT NUMERIQUE D'IMAGE.....	13
I.8CONCLUSION	15
I.1 INTRODUCTION	17
II.2.IMAGERIE PAR RESONANCE MAGNETIQUE (IRM) CEREBRALE.....	17
II.2.1 ANATOMIE CEREBRALE	17
II.2.2 PRINCIPE DE L'IMAGERIE PAR RESONANCE MAGNETIQUE (IRM)	18
II.2.3 FORMATION DES IRM	19
II.2.4 CONTRASTES EN IRM.....	20
II.2.5 CARACTERISTIQUES DES IRM CEREBRALES	20
II.2.5.1 LE BRUIT.....	20
II.2.5.2 L'EFFET DE VOLUME PARTIEL.....	21
II.2.5.3 LES HETEROGENEITES D'INTENSITE	21
II.2 DEFINITION DE LA SEGMENTATION.....	22
II.3 DIFFERENTES APPROCHES DE LA SEGMENTATION D'IMAGES	22
II.4.1APPROCHE FRONTIERE (CONTOUR).....	23
II.4.2 LES APPROCHE REGION	24
II.4.2.1 SEGMENTATION PAR CROISSANCE DE REGION (REGION GROWING)	24
II.4.2.2 SEGMENTATION PAR DIVISION/RASSEMBLEMENT (SPLIT AND MERGE).....	25
II.4.2.3 SEGMENTATION PAR CLASSIFICATION.....	26
II.4.2.3.1 METHODOLOGIE DE LA CLASSIFICATION.....	26

TABLE DES MATIERES

II.5 ALGORITHME DES K_MEANS.....	26
II.5 .1 PRINCIPE :	27
II.5.2 ALGORITHME DES CENTRES MOBILES (K-MEANS) :	27
II.5 LA CLASSIFICATION FLOUE.....	28
II.6.1 LA LOGIQUE FLOUE.....	28
II.6.2 LA THEORIE DES SOUS-ENSEMBLES FLOUS.....	28
II.7 CLASSIFICATION FLOUE FCM (FUZZY C-MEANS).....	28
II.7.1 DESCRIPTION DE L'ALGORITHME FCM.....	29
II.7.2 ALGORITHM FUZZY C-MEANS (FCM).....	30
II.8 COMPARAISON ENTRE K-MEANS ET FCM :	30
II.9 CONCLUSION :	31
III.1 LE TMS 320C6416 DSK.....	33
III.1.1 INTRODUCTION	33
III.1.2 LA FAMILLE TMS 320C.....	33
III.1.3 LES DIFFERENTES FAMILLES DE C6X.....	34
III.1.4 PRESENTATION DE LA PLATEFORME TMS 320C6000	34
III.1.5 CARACTERISTIQUES PRINCIPALES DU TMS 320C6416.....	35
III.1.6 ARCHITECTURE GENERALE DU STARTER KIT TMS 320C6416	36
III.1.6.2 STARTER KIT TMS320C6416 :	40
III.1.7 FONCTIONNEMENT :	41
III 2 CODE COMPOSER STUDIO	41
III 2 .1. INTRODUCTION.....	41
III 2.2. PRESENTATION DU CODE COMPOSER STUDIO CCS	41
III 2 .3. CONFIGURATION DU DEVELOPPEMENT ET ENVIRONNEMENT	43
III 2 .4. CREATION DU FICHIER EXECUTABLE	44
III 2 .5. LES FICHIERS SUPPORTS	45
III 2 .6. PROGRAMMATION A L'AIDE DU MATLAB/SIMULINK	46
III 2 .7. REAL-TIME WORKSHOP	47
III 2 .8. CONCLUSION.....	47
IV.1 INTRODUCTION.....	49
IV.2 PRESENTATION DU MODEL SIMULINK PROPOSE POUR L'IMPLEMENTATION.....	49
VI.2.1 PRESENTATION DU MODEL SIMULINK.....	49
IV. 3 IMPLEMENTATION DES ALGORITHMES SUR TMS320C6713 DSK.....	50
IV.4 SIMULATION ET RESULTATS OBTENUS	54
IV.4.1 INSTRUCTION POUR LA CONNEXION DE LA CARTE	54

TABLE DES MATIERES

IV.4.2 CREATION D'UN PROJET EN « C/C++ » A PARTIR D'UN MODEL SIMULINK.....	55
IV.4.4 DIFFERENTES OPERATIONS POUR L'IMPLEMENTATION SUR TMS320C6713 DSK	55
IV.4.5 RESULTATS OBTENU A L'AIDE DU CODE COMPOSER STUDIO CCS	57
CONCLUSION_	60
CONCLUSION GENERAL	63

LIST DES FIGURE

FIGURE I 1: REPRESENTATION ECHANTILLONNEE.....	4
FIGURE I 2EXEMPLE DE QUANTIFICATION SUR 8 BITS, 4 BITS, 2 BITS	6
FIGURE I 3CONSTRUCTION D'UNE IMAGE NUMERIQUE	6
FIGURE I 4 PIXEL « PICTURE ELEMENT ».....	7
FIGURE I 5 HISTOGRAMME DES NIVEAUX DE GRIS	9
FIGURE I 6 REPRESENTATION DE L'IMAGE A BASE RVB.....	11
FIGURE I 7 CODAGE EN NOIR ET BLANC	12
FIGURE I 8 CODAGE EN NIVEAUX DE GRIS.....	12
FIGURE I 9 DIFFERENTS OPERATIONS DE TRAITEMENT NUMERIQUE	14
FIGURE II 1 COUPE LONGITUDINALE DE L'HEMISPHERE DROIT	17
FIGURE II 2 COUPE FRONTALE DE L'ENCEPHALE [6].....	18
FIGURE II 3 – L'ETAT D'EXCITATION EST INSTABLE ET DES L'ARRET DE L'IMPULSION RF COMMENCE LE RETOUR A L'ETAT D'EQUILIBRE (STABLE).....	19
FIGURE II 4 FORMATION DE L'IMAGE : UN CHAMP DE GRADIENT PERMET DE SELECTIONNER UNE POSITION DANS UNE DIRECTION SPATIALE.....	20
FIGURE II 5 EFFET DE VOLUME PARTIEL D'APRES.....	21
FIGURE II 6 – ILLUSTRATION DE L'ARTEFACT D'INHOMOGENEITE. (A) CORRESPOND A L'IMAGE AFFECTEE PAR UNE HETEROGENEITE RF. (B) CORRESPOND `A L'ARTEFACT ISOLE ET (C) `A L'IMAGE CORRIGEE.....	22
FIGURE II 7 DIFFERENTS METHODES DE SEGMENTATION D'IMAGE	23
FIGURE II 8 RESULTAT DE LA SEGMENTATION PAR LA METHODE CROISSANCE DE REGION, (A): IMAGE ORIGINALE, (B): IMAGE BINAIRE, (C): IMAGE SEGMENTEE.....	24
FIGURE II 9 RESULTAT DE LA SEGMENTATION PAR LA METHODE SPLIT AND MERGE, (A): IMAGE INITIALE,	25
FIGURE III 1 LA FAMILLE TMS320C	34
FIGURE III 2 ARCHITECTURE DE TMS320C6416 DSP	37
FIGURE III 3 TMS320C6713 CPU DATA PATH	39
FIGURE III 4 LA CARTE STARTER KIT (C6416DSK).....	40
FIGURE III 5 SCHEMA GLOBALE DE LA CARTE C6416 DSK	40
FIGURE III 6 ORGANIGRAMME D'UN SYSTEME DE DEVELOPPEMENT DE LOGICIEL POUR DSP.....	42
FIGURE III 7 CAPTURE D'ECRAN DU CODE COMPOSER STUDIO.....	43
FIGURE III 8 CODE COMPOSER STUDIO.....	44
FIGURE III 9 CREATION DU FICHIER	44
FIGURE III 10 LES FICHIERS SUPPORTS	45
FIGURE III 11 LES FICHIERS SUPPORTS DANS LE CODE COMPOSER STUDIO	46

LIST DES FIGURE

FIGURE IV 1 ARCHITECTURE DU MODEL SIMULINK PROPOSE.....	50
FIGURE IV 2 TARGET PREFERENCES /C6416DSK.....	51
FIGURE IV 3 COLORSPACE CONVERSION	51
FIGURE IV 4 IMAGE DATA TYPE CONVERSION.....	52
FIGURE IV 5 IMAGE DATA TYPE CONVERSION.....	52
FIGURE IV 6 IMAGE DATA TYPE CONVERSION 1	53
FIGURE IV 7 VIDEO VIEWER.....	53
FIGURE IV 8 SINK BLOCK PARAMETERS :TUMOR SIZE	54
FIGURE IV 9 LE PASSAGE DU SIMULINK AU « C/C++ »	55
FIGURE IV 10 CREATION DU PROJET ALGORITHME_DETECTION_TUMEUR_CEREBRALE SUR CCS	56
FIGURE IV 11 RESULTATS OBTENU REELLEMENT AU LABO (INPUT IMAGE).....	57
FIGURE IV 12 RESULTATS OBTENU REELLEMENT AU LABO (OUTPUT IMAGE).....	58

LISTE DES TABLEAUX

TABLEAU I.1 COMPARATIF ENTRE TAILLES DES IMAGES	10
TABLEAU III.1 RESUME LES PRINCIPALES CARACTERISTIQUES DE DSP TEXAS INSTRUMENTS	37
TABLEAU III.2 COMPATIBILITE ENTRE MATLAB ET CCS	51
TABLE IV.1 DIFFERENTS BLOCS SIMULINK UTILISES	54
TABLE IV.2 LES DIFFERENTES IMAGES IRM AVANT ET APRES SEGMENTATION..	65

Abréviation et Acronymes

1D : image à une dimension

2D : image à deux dimensions

CAN : Conversion analogique Numérique

Bpp : bits par pixel

RVB : (R) rouge, (V) vert (Green en Anglais) , (B) bleu

LAN : Local area network

A/N : Analogique /Numérique

N/A : Numérique/ Analogique

IRM : Imagerie par résonance magnétique

MB : matière blanche

MG :matière grise

LCR : liquide céphalorachidien

GS :gradient de sélection de coupe

GP : gradient de codage de phase

GL :le gradient de lecture

T1 ou T2: Pondération T1 ou T2

TE: le temps d'echo

TR : le temps répétition

SEP : la sclérose en plaque

DSP : digital signal processors Transcranial magnetic stimulation,

DSK :TMS320C6416 DSP Starter Kit

GPS: The Global Positioning System (Le système de positionnement global)

RAS : Remote Access Service (Service d'accès distant)

DRAM : Dynamic random-access memory (Mémoire dynamique à accès aléatoire)

LEDs : Light-emitting diode (Diode électro-luminescente)

DIP switches : dual in-line package (double paquet en ligne)

CPLD : Complex Programmable Logic Device(Dispositif logique programmable complexe)

Abréviation et Acronymes

ALU: unités logiques arithmétiques

MIPS: millions d'instructions par seconde

MAC: multi-accumulations

MMACS: millions de MAC par Seconde

VCP: ViterbiDecoderCoprocesor

TCP: coproceseur de décodage de Turbo

McASP: multichannel audio serial port (portes série audio multicanales)

McBSP: multichannel buffered serial port (portes série protégées multicanales)

GPIO: General-Purpose- Input/Output (universel consacré d'entrée-sortie)

HPI: Host Port Interface (interface de centre-port)

EMIF: External Memory Interface (interface externe glueless mémoire)

IDE: Environnement de développement intégré

JTAG:Joint Team Action Group

Introduction générale

INTRODUCTION GENERALE

L'imagerie par résonance magnétique (IRM) a connu un véritable essor ces dernières années. Cette modalité d'imagerie est devenue un outil de plus en plus important en médecine du cerveau ou dans la recherche en neurosciences cognitives.

En effet, les champs d'exploration que cette technique offre sont larges: l'IRM anatomique qui permet d'observer avec une résolution fine les tissus cérébraux, l'IRM fonctionnelle qui offre la possibilité de visualiser l'activité cérébrale et l'IRM de diffusion qui permet d'explorer l'aspect de la connectivité des aires cérébrales. Pour diagnostiquer certaines maladies liées à des lésions cérébrales internes, le médecin doit analyser des images médicales. Pour étudier l'évolution d'une tumeur, il est nécessaire de connaître avec exactitude les changements survenus sur ces images.

L'interprétation visuelle des IRM cérébrales, n'est pas toujours sûre. C'est pour cela que le besoin d'une interprétation automatique qui permet d'assister les médecins dans leur prise de décision s'est fait ressentir.

Ainsi, pour une identification et un diagnostic fiables, dans le domaine médical, la précision est primordiale. En termes d'analyse d'images, il est plus que nécessaire que la segmentation soit précise. Les possibilités de traitement automatique de ces images s'avèrent pourtant délicates, car des capacités aussi banales pour l'oeil humain que la reconnaissance d'un objet posent de réelles difficultés pour l'outil informatique.

La segmentation d'images est un traitement de bas niveau, elle intervient au cœur de la discipline appelée « analyse ou traitement d'image ». C'est sans doute la clef de voûte de ce domaine, mais également la partie qui pose le plus de problèmes et qui s'avère la plus difficile à automatiser.

L'objectif de notre travail est de concevoir une méthode de segmentation d'images médicales, plus particulièrement des images IRM du cerveau. Notre méthode extrait les différentes principales matières composantes le cerveau (le cerveau, la tumeur...etc.).

INTRODUCTION GENERALE

Le premier chapitre est dédié au cadre d'application de ce mémoire. Il introduit les termes et concepts essentiels pour appréhender l'analyse et le traitement numérique d'image d'une façon générale.

Notre travail dans le chapitre deux présente quelques notions d'anatomie du cerveau, qui permettront d'introduire les principes de l'imagerie par résonance magnétique cérébrale. Ensuite les méthodes de segmentation d'images, où nous avons décrit un certain nombre de méthodes classiques de segmentation, soit par approche contour ou par approche région.

Dans le troisième chapitre on procédera à la description de la partie logicielle (l'environnement de développement intègre CCS) et la partie matérielle ce forme d'une étude générale sur le TMS 320 C 6416 DSK.

Le chapitre quatre est consacré à la présentation des résultats par l'algorithme que nous avons implémenté (la classification floue FCM). Nous exposons également les différents cas particuliers que nous avons rencontrés à l'issue de cette segmentation. Nous terminons ce travail par une conclusion générale.

Chapitre I :
Généralités sur le traitement
numérique d'image

I.1 Introduction

Dans ce chapitre, on se concentrera sur les concepts fondamentaux du traitement d'images. L'analyse d'image, sémantique ou non, nécessite des prérequis et des outils qui dépassent largement ceux de ce cours, tout en en reprenant certains aspects (par exemple l'extraction de primitives 1D ou 2D des images est nécessaire pour l'identification d'objets dans une scène). Par ailleurs, on ne traitera que les images 2D ici, mais la plupart des concepts abordés peuvent se généraliser à des images volumiques, voire de dimensions supérieures.

I.2 Image Numérique

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. [1]

I.2.1 Acquisition, Echantillonnage Spatial et Quantification:

L'acquisition d'images est une mesure spatiale d'une interaction entre une onde et de la matière. L'onde est émise par une source et reçue par un capteur. Par exemple dans le cas de l'échographie, l'ultrason, une onde acoustique, est émis et reçue par la sonde. L'interaction est la réflexion de l'ultrason sur la structure du corps. Dans le cas d'onde électromagnétique, la photographie utilise le spectre visible c'est-à-dire qui est visible pour l'œil humain. Il y a des applications sur l'ensemble du spectre électromagnétique, des rayons gamma jusqu'aux ondes radio. Ainsi, les images acquises par rayons X ou par rayons gamma sont surtout utilisées en imagerie médicale et en astronomie.

L'opération d'échantillonnage consiste à prélever sur un signal analogique dont l'évolution est continue dans le temps, des échantillons représentant l'amplitude aux instants de prélèvement. Pour des raisons de simplification, les prélèvements sont réalisés régulièrement avec une périodicité constante T_e appelée période d'échantillonnage. L'échantillonnage est qualifié d'idéal dès lors que l'on peut supposer ou approcher une prise instantanée des échantillons. [2]

Echantillonnage d'une fonction $f(x,y)$:

$$f_e(x,y) = f(x,y) \cdot \sum_i \sum_j \delta(x - i \Delta x, y - j \Delta y) \quad (1.1)$$

Δx est le pas d'échantillonnage dans la direction x

Δy est le pas d'échantillonnage dans la direction y

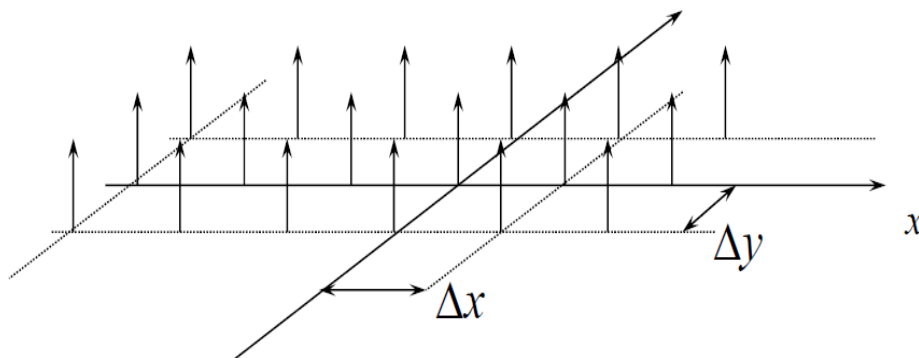


Figure I 1: Représentation échantillonnée

Tel que : $\sum_i \sum_j \delta(x - i\Delta x, y - j\Delta y)$ Peigne de Dirac 2D

Le poids $f(i\Delta x, j\Delta y)$ de chaque Dirac est :

- ✓ . Soit la valeur de $f(x,y)$ en $x = i \Delta x$ et $y = j \Delta y$
- ✓ . Soit la valeur «moyenne» de $f(x,y)$ dans une région entourant $(i \Delta x, j \Delta y)$ ($f(x,y)$ est pondérée et intégrée dans la région R).

Dans le cas général on aura (cas variant) :

$$\tilde{f}(i\Delta x, j\Delta y) = \int_R f(x, y)h(x, y, i\Delta x, j\Delta y)dx dy \tag{1.2}$$

Si $h(.,.)$ est identique en tout point (x,y) , on aura (cas invariant) :

$$\tilde{f}(i\Delta x, j\Delta y) = \int_R f(x, y)h(i\Delta x - x, j\Delta y - y)dx dy \tag{1.3}$$

h représentera la réponse impulsionnelle du système de prise de vue. C'est une opération de convolution, donc de filtrage.

L'image échantillonnée est donc :

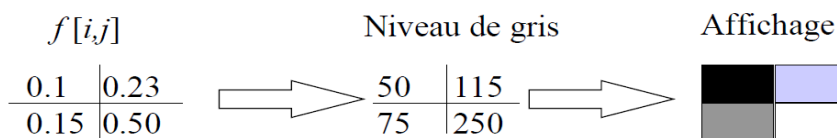
$$f_e(x, y) = \sum_i \sum_j \tilde{f}(i\Delta x, j\Delta y) \delta(x - i\Delta x, y - j\Delta y) \tag{1.4}$$

Dans un ordinateur, l'image (numérique) sera représentée par une matrice (tableau 2D) :

$$f[i, j] = \tilde{f}(i\Delta x, j\Delta y) \tag{1.5}$$

$f[i,j]$ est appelé : «valeur du PIXEL(i,j) » (Pixel: PICtureELement)

Pour visualiser une image, on remplit une région rectangulaire (Pixel) avec un niveau de gris (ou de couleur) correspondant à la valeur du pixel. En général les niveaux de gris (ou de couleur) utilisé pour la visualisation sont compris entre 0 et 255 (le code est de longueur fixe sur 8 bits).



- . CAN sur les systèmes d'acquisition d'images.
- . Codage de la valeur de chaque pixel sur N bits (En général 8 bits).

La quantification est la seconde étape nécessaire à la numérisation des signaux. Succédant à l'échantillonnage-blocage, elle permet le traitement numérique ou la mémorisation du signal. Son rôle est d'affecter une valeur de résolution finie à un échantillon dont l'amplitude est en théorie infiniment précise si l'on fait abstraction du bruit de fond propre au signal.

Quantifier un échantillon, c'est arrondir sa valeur à celle de l'échelon le plus proche sur une grille de niveaux. Lorsque les échelons sont à pas constant, la quantification est uniforme.

La quantification est une opération non conservatrice car la précision originale ne peut être retrouvée après arrondi. Elle peut être vue comme la superposition d'une composante aléatoire sur l'amplitude de chaque échantillon original.



Figure I 2 Exemple de quantification sur 8 bits, 4 bits, 2 bits

I.2.2 Construction d'une image numérique:

Une image numérique est composée d'unités élémentaires (appelé pixel) qui représentent chacun une portion de l'image. Une image est définie par :

- Le nombre de pixels qui la compose en largeur et en hauteur (qui peut varier presque à l'infini).
- L'étendu des teintes de gris ou des couleurs que peut prendre chaque pixel (on parle de dynamique de l'image).

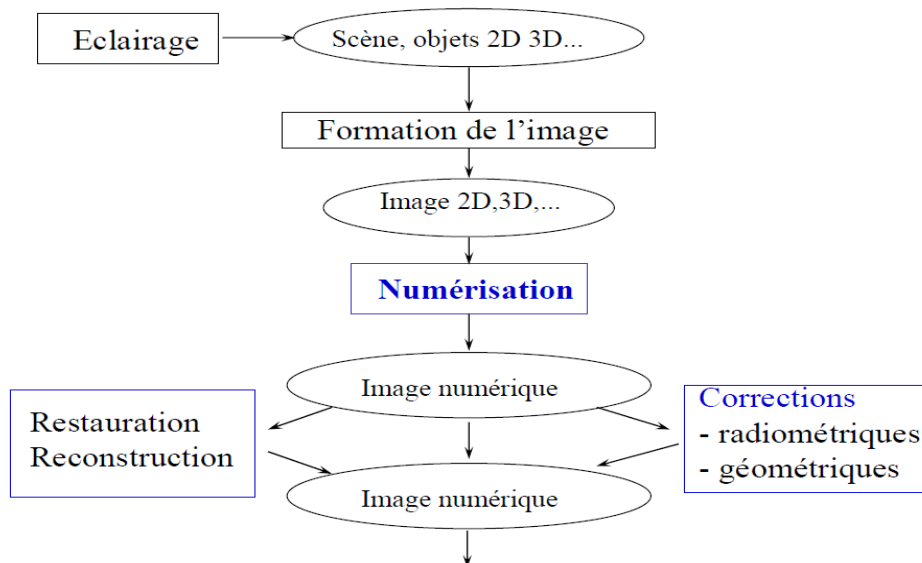


Figure I 3 Construction d'une image numérique

Toutes les données correspondant aux informations contenues dans l'image sont structurées d'une certaine façon afin de permettre leur stockage. Il existe un grand nombre de formats d'images, tous ces formats ne correspondent ni plus ni moins qu'à une structuration particulière des données concernant l'image. Une image numérique en elle-même est en fait un concept tout à fait abstrait (des données numériques) qui ne trouve une signification à nos yeux qu'à la visualisation lorsque l'on utilise un logiciel adéquat. La numérisation d'une image peut être obtenue par une simple Conversion analogique Numérique « CAN ».de la sortie du capteur. La synchronisation du balayage de l'image avec le CAN, permet d'échantillonner la fonction image en un nombre de points finis, dans le plan x-y et de mémoriser ainsi les valeurs obtenues. Les formats images usuelles sont : 32 *32 ; 100 *100 ; 256 *256 ; 512 *512. Une image peut être ainsi numérisée sous la forme d'un tableau à L lignes et C colonnes.[3]

I.3 Pixel

Pixel est l'élément d'image ou «Picture-élément », le pixel est le plus petit point de l'image numérique, c'est une entité calculable qui peut être quantité. Un pixel est aussi le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. L'ensemble des pixels est contenu dans un tableau à deux dimensions constituant l'image. [1]

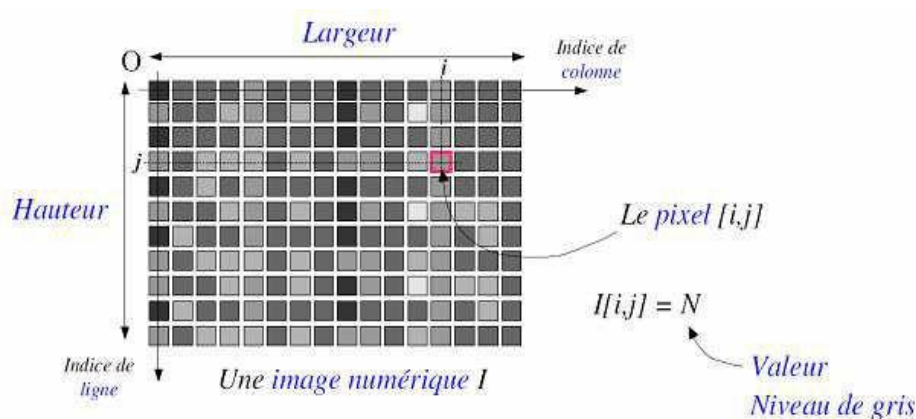


Figure I 4 Pixel « Picture élément »

Le codage des caractéristiques d'un pixel dépend du type d'images :

- **Images à deux niveaux** : Toute image décrite avec deux valeurs : une image en Noir/Blanc est l'exemple le plus courant. Un bit suffira ra pour coder la valeur d'un pixel (0 ou 1).
- **Images à plusieurs niveaux de gris** : Toute image décrite avec 2^n valeurs (n étant le nombre de bits de codage du pixel) : les images de télévision Noir/Blanc sont de ce type. La plus part des systèmes définissent plusieurs niveaux de gris maisseuls 128 (7bits) niveaux de gris sont détectables par l'œil .
- **Images couleurs** : La couleur peut être codée soit par composition de couleurs primaires, soit par composition d'informations de luminance et de chrominance.

Remarque : Si on prend comme exemple une simple image informatique type PC de taille 640 _ 480 pixels en vraies couleurs, soit 24bits par pixel (24bpp), la taille d'une telle image sera $640 \times 480 \times 24 = 7372800$ bpp ou 921; 600Ko (921; 600Ko).

C'est énorme ! Et le problème posé par une telle quantité d'information, intervient aussi bien pour la transmission, que pour le stockage des images. De là est né le besoin de compression des images afin de réduire le volume des données numériques.

I.4 Caractéristiques de l'image numérique

C'est la taille de l'image, qui n'est rien qu'une matrice de pixels. La taille est obtenue en multipliant le nombre de colonne par le nombre ligne. Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition (dimension) de 640 pixels par 480, notée 640 _ 480.

I.4.1 Taille d'une image

Pour connaître la taille (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. La taille (ou poids) de l'image est alors le nombre de pixels que multiplie la taille (en octets) de chacun de ces éléments

Voici le calcul pour une image 640 × 480 en Truecolor :

- Nombre de pixels : $640 \times 480 = 307200$ pixels / 8 = 3 octets
 - Le poids de l'image est ainsi égal à : $307200 \times 3 = 921600$ octets / 1024 = 900Ko
- Voici quelques exemples (en considérant que l'image n'est pas compressée) :

Finition de l'image	Noir et blanc (1bit)	256 couleurs (8bits)	65000 couleurs (16bits)	True color (24bits)
320x200	7.8Ko	62.5Ko	125Ko	187.5Ko
640x480	37.6Ko	300Ko	600Ko	900Ko
800x600	58.6Ko	468.7Ko	937.5Ko	1.4Mo
1024x768	96Ko	768Ko	1.5Mo	2.3Mo

Tableau I.1- Tableau comparatif entre tailles des images

Cela explique la mémoire vidéo que nécessite votre carte graphique en fonction de la résolution de l'écran (nombre de points affichés) et du nombre de couleurs. On voit sur l'exemple qu'il faut une carte ayant 4Mo de mémoire vidéo pour pouvoir afficher une résolution de 1024×768 en truecolor...

I.4.2. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'une image. La résolution est exprimée en unité de mesure (pouce ou centimètre) par nombre de pixels.

I.4.3 Histogramme

C'est une fonction qui donne la fréquence d'apparition de chaque niveau de gris ou couleur (Annexe B) dans l'image (fig1.5). Il peut être utile dans les cas suivant :

_ Pour diminuer l'erreur de quantification ;

- _ Pour comparer deux images obtenues sous des éclairages différents ;
- _ Pour améliorer certaine propriété d'une image (rehaussement) pour en tirer des informations utiles.

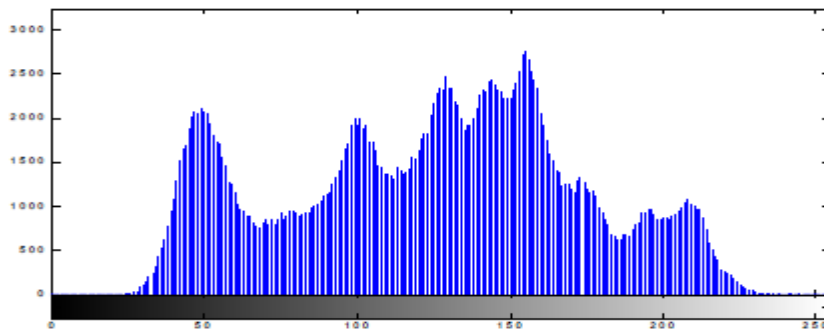


Figure I 5 Histogramme des niveaux de gris

I.4.4 Luminance

C'est la variation d'intensité lumineuse ou le degré de luminosité des points de l'image, une bonne luminance se caractérise par :

- _ Des images brillante
- _ Un bon contraste
- _ L'absence de parasites

I.4.5 Niveau de gris

La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires, on peut donc dire que chaque pixel de l'image correspond à une quantité de lumière renvoyé comprise entre 0 et 255.

I.4.6 Bruit et dégradation

Avant de s'intéresser aux améliorations à apporter à l'image numérique pour faciliter leur interprétation, il est important de comprendre quels types de dégradations peuvent intervenir sur les images. Il existe deux types de dégradations : les perturbations aléatoires (bruit) et les perturbations déterministes (comme le flou). Ces perturbations sont souvent identifiées et formalisées afin de pouvoir prédire et donc réduire leur effet sur le signal capturé. L'origine des perturbations est aussi diverse qu'il existe d'outils et d'environnements d'acquisition différents. Il n'existe donc pas qu'un seul type de distorsion.

Les premiers bruits identifiés sont dus au matériel d'acquisition. Un capteur a une sensibilité minimum, et si la luminosité ambiante se situe en dessous de ce seuil, des détails sont perdus. Afin de résoudre ce problème, la solution la plus valable est d'augmenter la luminosité de la scène, en prolongeant la durée d'exposition ou encore en ouvrant un peu plus l'objectif, le capteur reçoit plus de lumière et passe le seuil minimal de rendement.

À l'inverse, pour les signaux ayant une amplitude trop élevée, on observe un phénomène d'éblouissement. Les condensateurs emmagasinent une charge excessive qui a alors tendance à s'écouler dans les cellules voisines, créant ainsi des effets de halo ou de traînée. Ces effets étant difficiles à réduire après acquisition, la solution la plus souvent envisagée est de

chercher à réduire l'effet visuel en amont de la prise de vue, aussi bien pour les capteurs argentiques (couche antihalo) que pour les capteurs numériques (isolation des cellules).

En plus de ces bruits inhérents aux technologies utilisées pour les capteurs, il y a une multitude d'autres sources de perturbations. Certaines sont négligeables, comme le bruit photonique (bruit grenaille) qui est dû à l'arrivée aléatoire des photons, tandis que d'autres gommement fortement les détails d'une image. C'est le cas du flou. Cette distorsion est souvent dû à une mauvaise mise au point, mais aussi, parfois, à une surface qui diffuse la lumière (la buée par exemple). Il peut être aussi causé par le mouvement d'un objet ou de toute la scène. On parle alors de flou de mouvement. Ces deux flous ne sont pas modélisés de la même manière.

Dans le premier cas, le flou est dit bidirectionnel et est approximé par un filtre moyenneur bidimensionnel (souvent gaussien) tandis que dans le second cas, le filtre doit tenir compte d'une estimation du mouvement pour adapter la répartition des coefficients en fonction de la direction et de l'intensité du mouvement.

Enfin, il existe une autre sorte de perturbation, dû au support lui-même. Le temps ou des événements indirects (une tache de café, le papier froissé) provoquent des perturbations dont la modélisation par une loi de probabilité est plus complexe et nécessite l'utilisation d'outils adaptatif, c'est-à-dire fonction de paramètres locaux à l'image. La modélisation n'est plus fonction d'une loi générale simple et nécessite l'utilisation d'outils comme les champs de Markov pour prendre en compte d'autres paramètres spatiaux. [1]

I.4.7 La transparence de l'image

La transparence est une caractéristique permettant de définir le niveau d'opacité des éléments d'une image, c'est-à-dire la possibilité de voir à travers l'image des éléments graphiques situés derrière celle-ci.

Il existe deux modes de transparence :

- La transparence simple s'applique pour une image indexée et consiste à définir parmi la palette de couleurs une des couleurs comme transparente.
- La transparence par couche alpha (ou canal alpha, en anglais alpha Channel) consiste à rajouter pour chaque pixel de l'image un octet définissant le niveau de transparence (de 0 à 255). Le processus consistant à ajouter une couche transparente à une image est généralement appelé alpha blending .

I.4.8 Image en noir et blanc et image en couleurs

Image en noir et blanc ou image monochrome est caractérisée par des niveaux de gris. Le niveau de gris recherché va correspondre à la valeur de l'intensité lumineuse recherchée.

Ici, chaque pixel est codé sur un octet (8bits) ce qui donne une valeur comprise entre 0 et 255 pour ce pixel. On a donc 2^m niveaux de gris où m est le nombre de bits du pixel. Si cette valeur du pixel est proche de 0 alors on aura une couleur noire. Si la valeur est proche de $2^m - 1$ alors on aura du blanc.

Images en couleurs est obtenue par superposition de trois rayonnements lumineux : le rouge (R), le vert (V, Green en Anglais) et le bleu (B). Dans le cas d'un écran cathodique, ces 3 rayonnements sont obtenus en bombardant les luminophores photo-sensibles de l'écran.

Mais il existe plusieurs standards de représentation de couleur ont nous allons citer quelques-uns dans le paragraphe qui va suivre. [4]

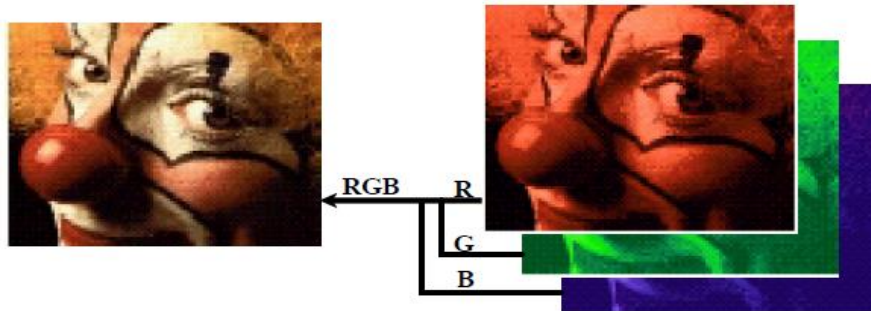


Figure I 6 Représentation de l'image à base RVB

I.5 Les différents types d'image numérique

Il existe 2 sortes d'images numériques : les images matricielles et les images vectorielles.

I.5.1 Image matricielle

Elle est formée d'une grille de points ou pixels, chacun pouvant avoir une couleur différente. Une image matricielle est caractérisée notamment par sa dimension en pixels, sa résolution et son mode colorimétrique. Les images vues sur un écran de télévision ou une photographie sont des images matricielles. On obtient également des images matricielles à l'aide d'un appareil photo numérique, d'une caméra vidéo numérique ou d'un scanner.

On représente souvent cette matrice numérique sous forme d'un tableau. Matrice S sous forme d'un tableau Cette disposition a l'avantage de respecter les voisinages spatiaux des pixels. Les pixels sont ainsi numérotés en fonction de leur position (numéro de ligne, numéro de colonne).

$$S = \begin{bmatrix} s(1,1) & s(1,2) & s(1,3) & \dots & \dots \\ s(2,1) & s(2,2) & s(2,3) & \dots & \dots \\ s(3,1) & s(3,2) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & s(i,j) & \dots & \dots \end{bmatrix}$$

I.5.2 Image vectorielle

Elle n'est pas composée de pixels mais définie par des fonctions mathématiques qui décrivent des lignes, des courbes etc. Dans ce cas on manipule des objets et non des pixels. Par exemple, un cercle est décrit par une fonction du type (cercle, position du centre, rayon). Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans. Ces images présentent 2 avantages : elles occupent peu de place en mémoire et peuvent être redimensionnées sans perte d'information.

I.6 Codage d'une image numérique

I.6.1 Codage de l'information

L'information est codée en binaire. Le support évolue mais le principe est toujours le même, un même élément peut se trouver dans 2 états différents stables. Il constitue une mémoire élémentaire ou bit. Conventionnellement on attribue le symbole 0 à l'un de ces 2 états et le symbole 1 à l'autre [2]

I.6.2 Codage d'une image binaire

Pour ce type de codage, chaque pixel est soit noir, soit blanc. Il faut un bit pour coder un pixel (0 pour noir, 1 pour blanc). L'image de 10000 pixels codée occupe donc 10000 bits en mémoire. Ce type de codage peut convenir pour un plan ou un texte mais on voit ses limites lorsqu'il s'agit d'une photographie dans la figure suivante.



Figure I 7 Codage en noir et blanc

I.6.3 Codage d'une image en niveaux de gris

Si on code chaque pixel sur 2 bits on aura 4 possibilités (noir, gris foncé, gris clair, blanc). L'image codée sera très peu nuancée. En général on code chaque pixel sur 8 bits = 1 octet. On a alors 256 possibilités (on dit 256 niveaux de gris). L'image de 10000 pixels codée occupe alors 10000 octets en mémoire.



Figure I 8 Codage en niveaux de gris

I.6.4 Codage d'une image en couleurs 8 bits

Dans ce cas on attache une palette de 256 couleurs à l'image. Ces 256 couleurs sont choisies parmi les 16 millions de couleurs de la palette RVB. Pour chaque image le programme recherche les 256 couleurs les plus pertinentes. Chaque code (de 0 à 255) désigne une couleur. L'image occupe 3 fois moins de place en mémoire qu'avec un codage 24 bits. L'image est moins nuancée : sa qualité est bonne mais moindre.

I.6.5 Codage d'une image en couleurs 24 bits

Il existe plusieurs modes de codage de la couleur. Le plus utilisé est le codage Rouge, Vert, Bleu (RVB). Chaque couleur est codée sur 1 octet = 8 bits. Chaque pixel sur 3 octets c'est à dire 24 bits : le rouge de 0 à 255, le vert de 0 à 255, le Bleu de 0 à 255.

Le principe repose sur la synthèse additive des couleurs : on peut obtenir une couleur quelconque par addition de ces 3 couleurs primaires en proportions convenables. On obtient ainsi $256 \times 256 \times 256 = 16777216$ (plus de 16 millions de couleurs différentes).

L'image de 10000 pixels ainsi codée occupe $10000 \times 3 = 30000$ octets. C'est ce codage de la couleur qui est utilisé par la plupart des écrans d'ordinateurs actuellement. On constate qu'il est très gourmand en mémoire. Pour faciliter le stockage des images en mémoire on utilise d'autres formes de codage.

I.7 Traitement Numérique d'image

I.7.1 Définition

C'est un ensemble des opérations relatives à la collecte, à l'enregistrement, à l'élaboration, à la modification, à l'édition, . . . des données.

Mettons de cotés les termes enregistrement et édition. Le principe général du traitement d'image est donc à quelques détails près toujours le même un système reçoit des images, y applique un traitement, et produit une information de nature liée à l'application visée.

Une source de rayonnement envoie des ondes sur un objet, qui sont ensuite réfléchies et collectées par un capteur. Le capteur transforme ces ondes en un ensemble de points. Ces points sont traités et une information est produite en sortie du système. On peut résumer le traitement d'image en quatre étapes principales : [3]

I.7.1.1 Acquisition des images

Mise en œuvre des processus physiques de formation des images suivis d'une mise en forme pour que ces images puissent être traitées par des systèmes informatiques.

I.7.1.2 Traitement des Images

Son but : améliorer ces images lorsqu'elles possèdent du bruit ou des défauts.

I.7.1.3 Segmentation des images

Son but : construire une image symbolique en générant des régions homogènes selon un critère défini à priori.

I.7.1.4 Analyse des images

Consiste à extraire des paramètres ou des fonctions représentatives de l'image ou des régions.

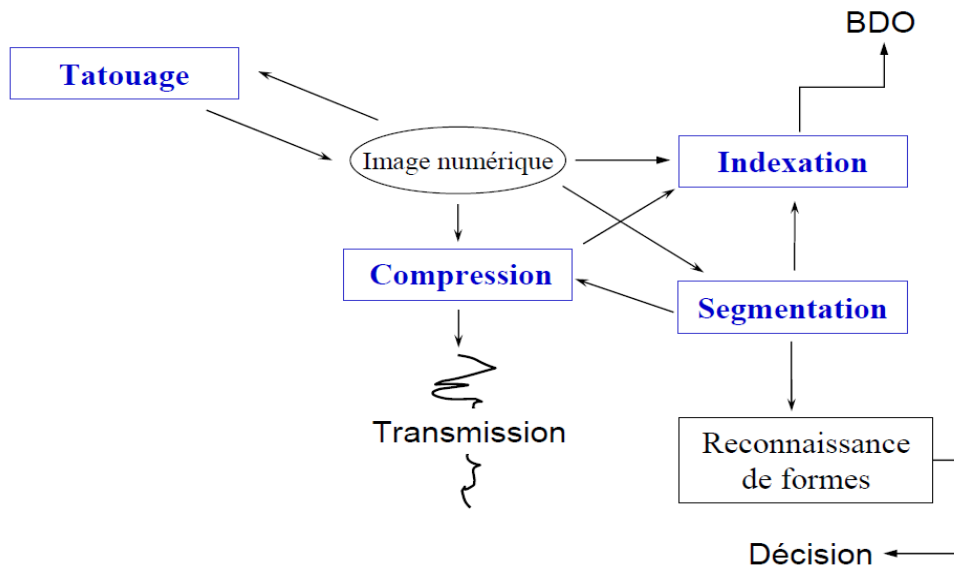


Figure I 9 Différents opérations de traitement Numérique

I.7.2 Les Avantages de traitement numérique

(Comparativement au traitement analogique) qu'on peut les citer sont les suivants : [5]

✓ flexibilité :

Il est possible de reconfigurer ou de reprogrammer un système de traitement numérique, simplement échangeant des paramètres ou des coefficients dans un programme. Au contraire, modifier un système de traitement analogique du signal requiert de changer des composants physiques, ce qui inclut des procédures de design et de validation.

✓ Précision :

Une bonne précision peut être atteinte par le traitement numérique. Cette précision dépend de la résolution numérique du système numérique utilisé (16 bits, 32, bits), et de la résolution des convertisseurs A/N et N/A qui sont utilisés (8 bits, 12bits, 16 bits ou 18 bits). Par exemple, un filtre numérique ayant des caractéristiques fréquentielles quasi-idéales est facile à réaliser en temps discret. Au contraire, pour les systèmes de traitement analogique du signal, l'imprécision sur les composants (par exemple sur des résistances et des condensateurs : 1% ou 5%) fait qu'il n'est pas possible d'avoir des filtres analogiques quasi-idéaux.

✓ Capacité de réaliser des systèmes plus complexes :

Réaliser un système numérique ayant une réponse désirée complexe est relativement facile, puisqu'il s'agit de trouver les bons coefficients et de les utiliser ensuite dans un programme ou dans un circuit numérique. Au contraire, réaliser une certaine réponse désirée complexe avec un système analogique en temps continu peut être beaucoup plus difficile. Certains algorithmes sophistiqués peuvent difficilement se mettre en œuvre en temps continu, et peuvent facilement se mettre en œuvre en temps discret. Le traitement numérique du signal a donc permis de réaliser des traitements qui n'étaient pas envisageables jusqu'à présent.

✓ Mémorisation et entreposage de données :

Les médias numériques (tels disques durs, CD-ROM, etc.) qui sont couramment utilisés par les ordinateurs modernes sont faits pour enregistrer l'information de signaux numériques et en temps discret, et non de signaux analogiques et en temps continu.

✓ **Transmission de données :**

Les réseaux de communication modernes (Internet, LAN, etc.) sont également conçus pour transmettre des données numériques et en temps discret.

✓ **Coût :**

Le coût des systèmes de traitement numérique diminue de plus en plus, à cause de la baisse des prix des circuits électroniques numériques. De plus, à cause du gain de flexibilité des systèmes, le coût de ces systèmes est parfois inférieur au coût d'un système de traitement analogique.

Ces avantages font en sorte que le traitement numérique est utilisé depuis 30 ans dans plusieurs domaines, dont certains sont :

- Traitement de la parole, de la musique et des images.
- Égalisation numérique de lignes téléphoniques et de canaux de télécommunications.
- Détection de séismes, de couches de pétrole, d'explosions nucléaires.
- Contrôle numérique adaptatif.

I.8 Conclusion

Une fois l'image numérique entièrement traitée par ordinateur, il faut, pour la visualiser à l'écran et donc en profiter sur papier ou sur un autre type de support (écran), la reconstituer en signal analogique à l'aide de restaurateurs. Il convient donc de recréer l'image de départ à l'aide d'une multitude de fragments de pixels de l'image numérique en leur donnant une forme la plus proche possible de l'image d'origine et en tenant compte des erreurs et parasites éventuellement rencontrés.

Chapitre II :
Segmentation d'image IRM
Cérébrale

II.1 Introduction

La segmentation d'images constitue une étape essentielle en traitement d'images. De nombreuses méthodes ont été proposées dans la littérature, dont la plus part sont basées sur le seuillage. Dans cette partie, nous présenterons les différentes approches de la segmentation d'images, ensuite, nous nous intéresserons aux méthodes de classification supervisée en général et la classification non supervisée en particulier.

II.2. Imagerie par Résonance Magnétique (IRM) cérébrale

II.2.1 Anatomie cérébrale

L'encéphale est constitué du cerveau, du cervelet et du tronc cérébral. Le cerveau est composé de trois matières principales : la matière blanche (MB), la matière grise (MG) et le liquide céphalorachidien (LCR). La matière blanche est constituée des fibres des cellules nerveuses appelées axones qui permettent la transmission de l'information traitée au niveau de la matière grise. La matière grise contient le corps des cellules nerveuses et est répartie en deux types de structures : le cortex et les noyaux. Le cortex est caractérisé par de nombreuses fissures appelées sillons. Sur la face intérieure du cortex se trouve la matière blanche et sur la face extérieure circule la matière grise. Les noyaux, constitués essentiellement de matière grise, sont des structures plus compactes au centre du cerveau. Le liquide céphalo-rachidien baigne la surface extérieure du cerveau et du cervelet et remplit le système ventriculaire. Les figures II.1 et II.2 illustrent cette description de l'anatomie cérébrale.

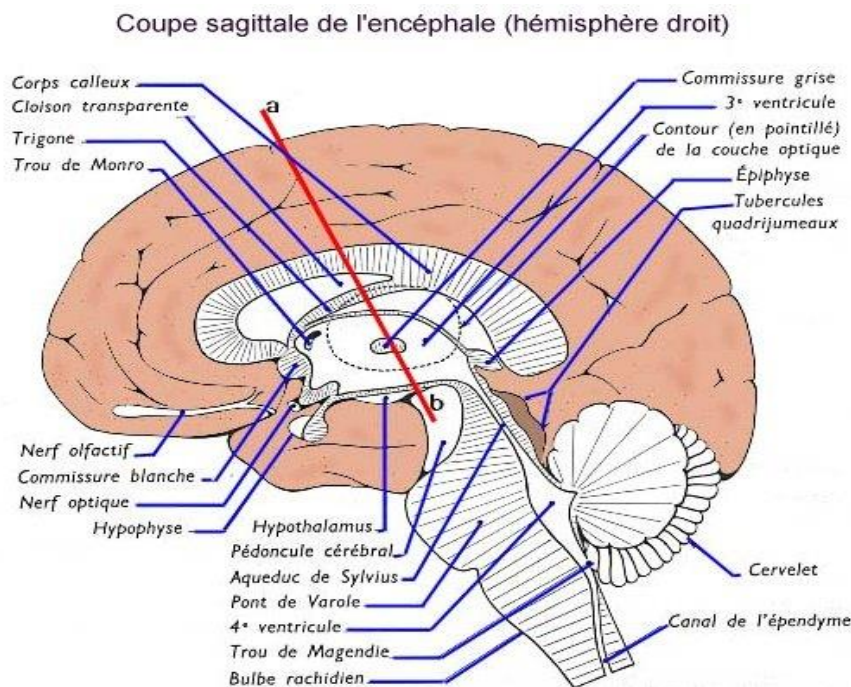


Figure II 1 Coupe longitudinale de l'hémisphère droit

II.2.2 Principe de l'Imagerie par Résonance Magnétique (IRM)

L'IRM est une technique de diagnostic médical fournissant des images tridimensionnelles et en coupe de grande précision anatomique. Cette technique récente, non invasive, est basée sur le phénomène physique de la résonance magnétique nucléaire. La résonance magnétique nucléaire est une technique en développement depuis une soixantaine d'années dont le phénomène physique a été conceptualisé en 1946 par Bloch et Purcell. Les premiers développements en Imagerie par Résonance Magnétique datent des années 1973. Les premières images chez l'homme ont été réalisées en 1979.

Aujourd'hui, l'IRM est devenue une technique majeure de l'imagerie médicale moderne. Le principe de l'IRM repose sur la propriété de certains atomes à entrer en résonance dans certaines conditions :

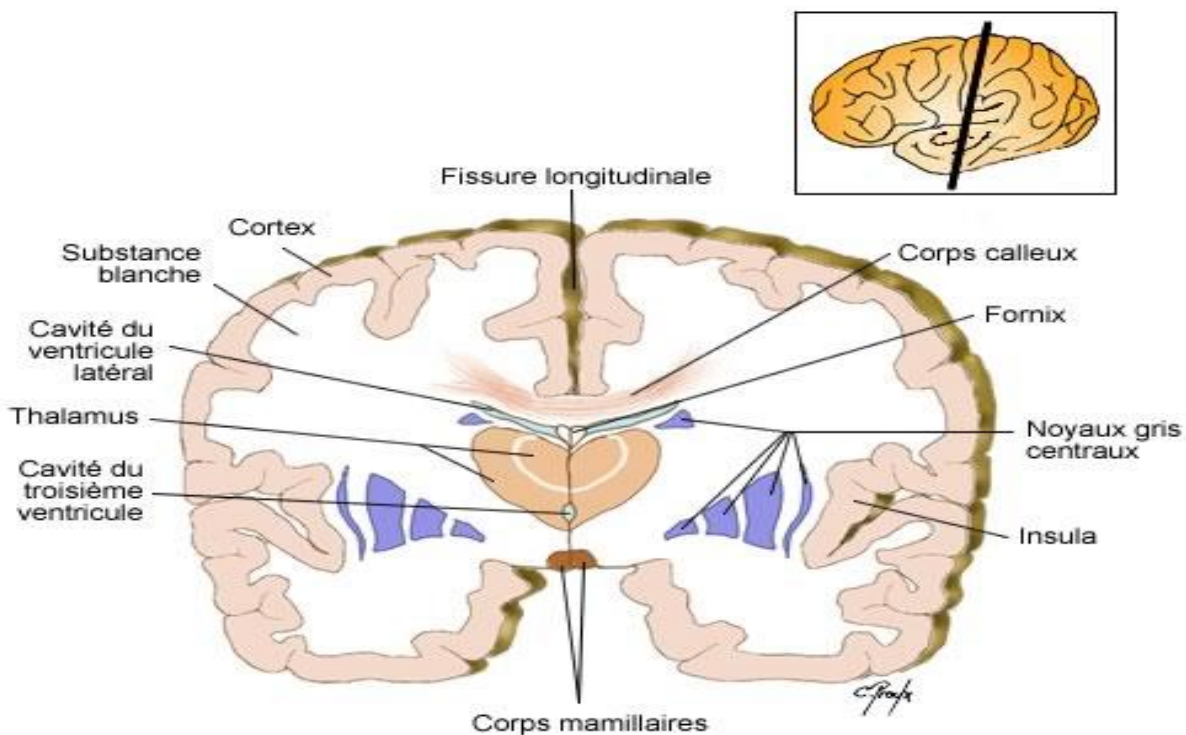


Figure II 2 Coupe frontale de l'encéphale [6]

C'est le cas de l'atome d'hydrogène (H) qui est un des deux constituants de la molécule d'eau (H_2O) que l'on trouve en grande quantité dans le corps humain (entre 60 et 75%). De plus, la quantité d'eau varie d'un tissu à l'autre, mais aussi à l'intérieur d'un même tissu selon son état physiologique, ce qui est utilisé pour établir une cartographie du corps humain et de ses pathologies.

Lorsque les atomes d'hydrogène entrent en résonance sous l'action d'un champ magnétique, ils absorbent de l'énergie : ainsi, plus la concentration en eau du milieu est élevée, plus il y a absorption d'énergie. A l'arrêt du phénomène de résonance, l'énergie emmagasinée par les atomes est restituée au milieu.

II.2.3 Formation des IRM

Le processus se fait en trois étapes : dans un premier temps, le corps est placé dans un champ magnétique qui oriente tous les protons dans la même direction. Puis les protons sont excités par des ondes radio qui modifient leur orientation. Dès l'arrêt de l'impulsion RF (Radiofréquence), les protons retournent à l'état d'équilibre. L'acquisition des signaux RMN émis permet de reconstruire l'image.

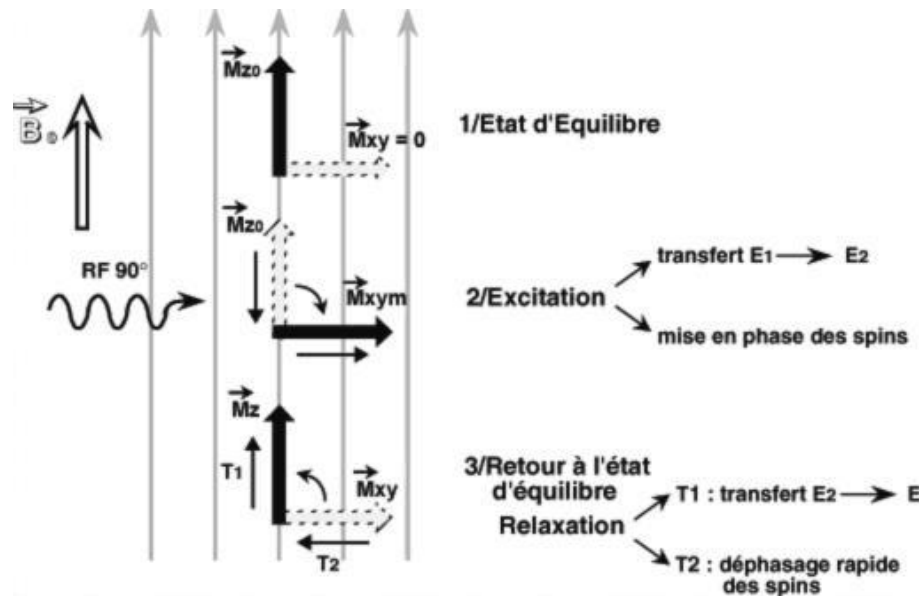


Figure II 3 – L'état d'excitation est instable et dès l'arrêt de l'impulsion RF commence le retour à l'état d'équilibre (stable).

Pour former le volume, il faut pouvoir localiser dans les trois plans de l'espace le signal reçu par l'antenne réceptrice. Pour cela trois gradients de champs magnétiques sont utilisés. Le gradient de sélection de coupe (GS) est d'abord appliqué puis le gradient de codage de phase (GP) permet de sélectionner les lignes de la matrice à l'origine de l'image et le gradient de lecture (GL) permet de sélectionner les colonnes (Fig. II.4). Une transformée de Fourier permet ensuite de reconstruire l'image.

Lors de l'acquisition du signal, le processus d'échantillonnage engendre un phénomène appelé volume partiel : certains voxels issus de l'échantillonnage du volume observé sont à la frontière entre deux tissus et peuvent donc contenir de l'information de plusieurs classes tissulaires. Ces voxels appartiennent donc à plusieurs classes en même temps, du fait de la résolution spatiale de l'imageur, ce qui pose un problème lors de la labellisation.

Les images IRM sont donc obtenues par application de gradients de champ et de séquences d'impulsions RF.

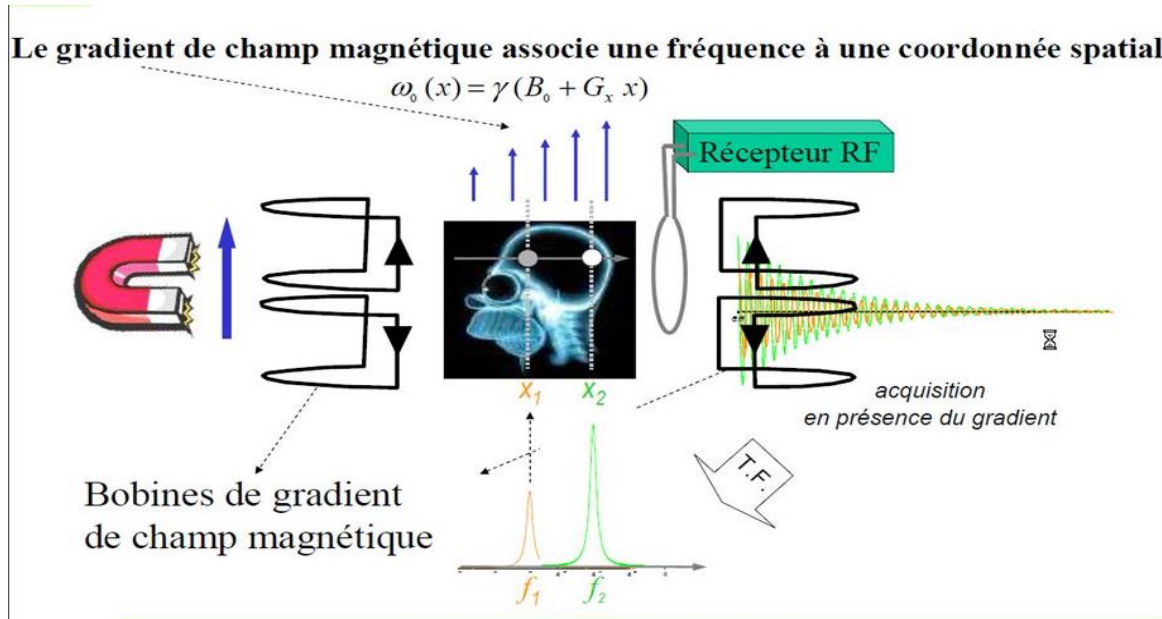


Figure II 4 Formation de l'image : un champ de gradient permet de sélectionner une position dans une direction spatiale

II.2.4 Contrastes en IRM

Après avoir vu les notions de base concernant l'imagerie par résonance magnétique, nous allons maintenant décrire les différents types de contrastes qui permettent d'obtenir des images contenant des informations de natures différentes nommées modalités. On peut ainsi pondérer l'image en T1 ou T2, ou en densité de protons. Les différents types de contraste sont obtenus en faisant varier les paramètres d'acquisition : le temps d'écho TE, c'est-à-dire le temps qui sépare l'impulsion RF et l'acquisition du signal, et le temps de répétition TR, c'est-à-dire le temps entre deux impulsions RF consécutives. Les images pondérées en T1 sont obtenues lorsque le TR et le TE sont courts tandis que les images sont pondérées en T2 lorsque le TR et le TE sont longs. Pour les images pondérées en densité de protons, le contraste est obtenu en utilisant un TR long et un TE court.

Chaque modalité contient des informations spécifiques ne se retrouvant pas dans les autres. A partir des images pondérées en T1, on peut par exemple distinguer les différents tissus cérébraux : matière blanche, matière grise et liquide céphalo-rachidien ; tandis que les images pondérées en T2 mettent plus facilement en évidence certaines anomalies comme les lésions dues à la sclérose en plaque (SEP).

II.2.5 Caractéristiques des IRM cérébrales

II.2.5.1 Le bruit

Un bruit aléatoire gaussien apparaît dans le domaine de Fourier de l'image acquise. L'image est ensuite obtenue en calculant le module de la transformée inverse de Fourier. La distribution gaussienne devient donc, dans l'image, une distribution de Rice [7]. Dans les

zones où l'intensité n'est pas proche de zéro, cette distribution est approximativement gaussienne, tandis que dans les zones où l'intensité est proche de zéro, elle est proche d'une distribution de Raleigh.

Dans la suite du manuscrit, nous ferons l'hypothèse d'une distribution gaussienne.

II.2.5.2 L'effet de volume partiel

Un autre problème de la segmentation d'images IRM est l'effet de volume partiel qui apparaît lorsque plusieurs types de tissus contribuent au même voxel. Ce problème est de plus en plus pris en compte dans les algorithmes de segmentation. En raison d'une résolution du système d'acquisition limitée, les voxels situés à la frontière entre plusieurs tissus sont composés de deux ou plusieurs tissus. Il est donc nécessaire de prendre en compte ces effets de volume partiel pour obtenir une segmentation fiable des tissus cérébraux. Les méthodes de classification en 3 classes "dures" (matière blanche, matière grise et liquide céphalorachidien) ignorent ce problème et perdent ainsi de l'information sur la structure des tissus.

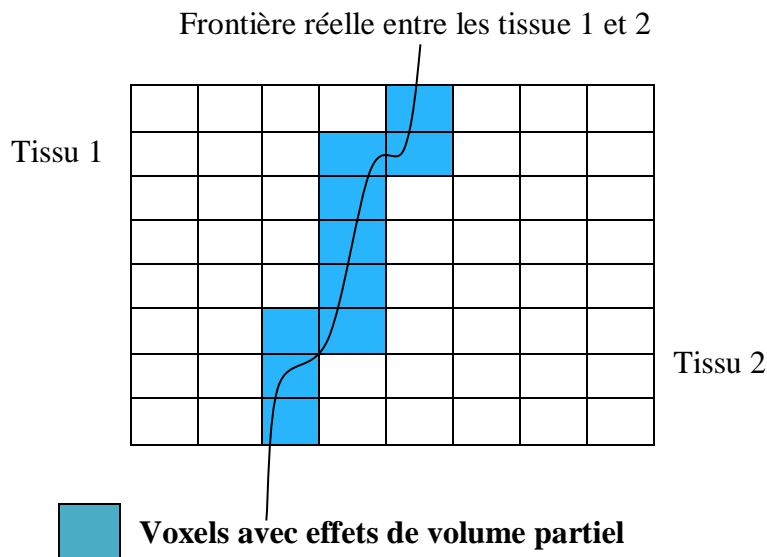


Figure II 5 Effet de volume partiel d'après

Certaines méthodes considèrent l'effet de volume partiel comme un facteur de dégradation et cherchent à le corriger, alors que d'autres le considèrent comme une propriété de l'image et cherchent à l'estimer pour obtenir une qualité sous-voxel [8].

II.2.5.3 Les hétérogénéités d'intensité

Une des principales difficultés de la segmentation d'images IRM est la présence d'un artefact d'hétérogénéité d'intensité spatiale pour un même tissu cérébral. Les inhomogénéités du champ RF sont en effet responsables de ces variations spatiales lentes de l'intensité des images (Fig. II.6). Ce biais peut poser des problèmes de classification pour des techniques de segmentation basées sur l'intensité, si on suppose que l'intensité d'une classe est constante sur toute l'image.

La non-uniformité est prise en compte dans la plupart des méthodes de segmentation, soit en la compensant par prétraitement [9], soit en la modélisant au cours de la segmentation. [10].

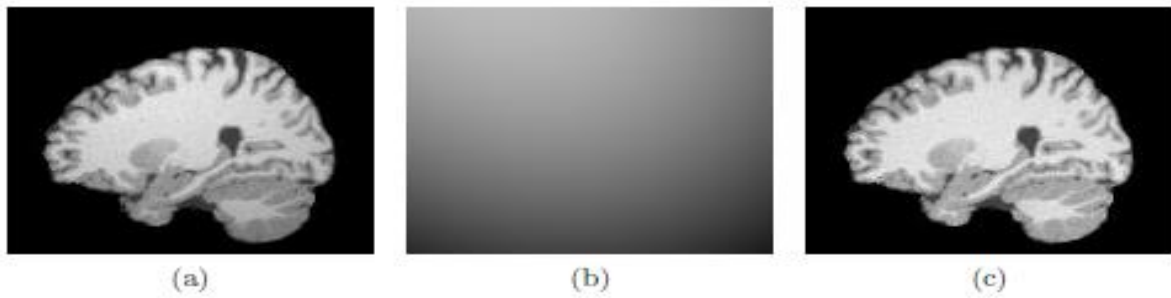


Figure II 6 – Illustration de l'artefact d'inhomogénéité. (a) correspond à l'image affectée par une hétérogénéité RF. (b) correspond à l'artefact isolé et (c) à l'image corrigée.

II.2 Définition de la segmentation

La segmentation d'images est un traitement de bas-niveau et une des étapes critiques de l'analyse d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis ainsi c'est un domaine vaste où l'on retrouve de très nombreuses approches. Toutes ces approches visent à l'extraction des indices visuels. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. On peut même montrer que le problème de la segmentation est le plus souvent un problème mal posé. Etant donnée une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu. Entre autres, elle devra éviter les choix irréversibles.

De façon plus précise, on peut définir la segmentation comme étant une partition d'une image I en n ensembles R_i appelés régions tels que :

$$\bigcup_{i=1}^n R_i = I \quad \text{avec} \quad R_i \cap R_j = \emptyset \quad i \neq j$$

et

$$\begin{cases} P(R_i) = \text{vrai} & \forall i = 1, \dots, n \\ P(R_i \cup R_j) = \text{faux} & R_i \text{ adjacente à } R_j \end{cases}$$

II.3 Différentes Approches de la segmentation d'images

À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en deux principales classes :

- Segmentation fondée sur les régions (en anglais : region-based segmentation). On y trouve par exemple : la croissance de région (en anglais : region-growing), décomposition/fusion (en anglais : split and merge).
- Segmentation fondée sur les contours (en anglais : edge-based segmentation).

Le schéma de la figure (1.8) nous donne une taxinomie de ces différentes approches.

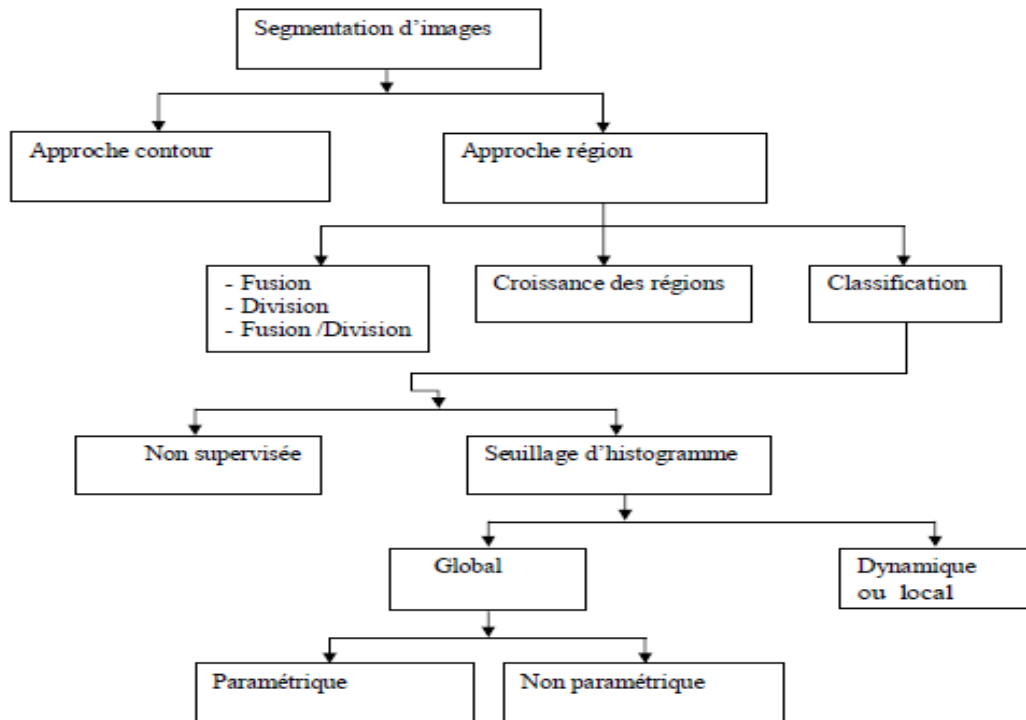


Figure II 7 Différents méthodes de segmentation d'image

II.4 Différentes Approches de la segmentation d'images

II.4.1 Approche frontière (contour)

Dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes ; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensités présentant un maximum ou un minimum ; il s'agit alors de contour « en toit ».

La notion de « frontière » est associée à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles connexes de points [7].

Donc l'approche locale de détection de contour consiste à balayer l'image avec une fenêtre définissant la zone d'intérêt. A chaque position, un opérateur est appliqué sur les pixels de la fenêtre afin d'estimer s'il y a une transition significative au niveau de l'attribut choisi. A partir des pixels susceptibles d'appartenir à un contour, il faut ensuite extraire des contours fermés [8].

II.4.2 Les approche région

L'approche "région" de la segmentation utilise des techniques d'identification et de localisation d'ensembles connexes de pixels. Les méthodes par classification ont pour but de partitionner les images en plusieurs classes - comme leur nom l'indique - et constituent le plus souvent une étape dans la segmentation d'objet `a proprement parler. Cependant, leur utilisation dans les méthodes de segmentation étant très répandue, il nous a paru judicieux d'en expliquer les ressorts. Ces différentes classifications peuvent être séparées selon plusieurs critères : probabilistes ou déterministes, paramétriques ou non, supervisées ou non. Nous présenterons donc un panel de ces méthodes parmi lesquelles nous trouverons les réseaux de neurones (méthode déterministe supervisée), les approches K-Moyennes, Fuzzy C-Means, Adaptive Fuzzy C-Means et Mean Shift (déterministes non supervisées), les différentes mixtures de lois (probabilistes paramétriques) et les approches markoviennes et par Machine `a Vecteurs de Support (probabilistes non-paramétriques).

II.4.2.1 Segmentation par croissance de région (région growing)

La méthode de segmentation en utilisant la croissance de région est résumée comme suit :

- La méthode de croissance de régions est une méthode de fusion.
- L'image est décomposée en primitives 'régions' (une région \equiv un seul pixel).
- Celles-ci sont ensuite regroupées de manière itérative selon un ou plusieurs critères de similarité, jusqu'à ce qu'il n'y ait plus de fusion possible.

L'Application de la méthode de croissance de région sur une image IRM est :

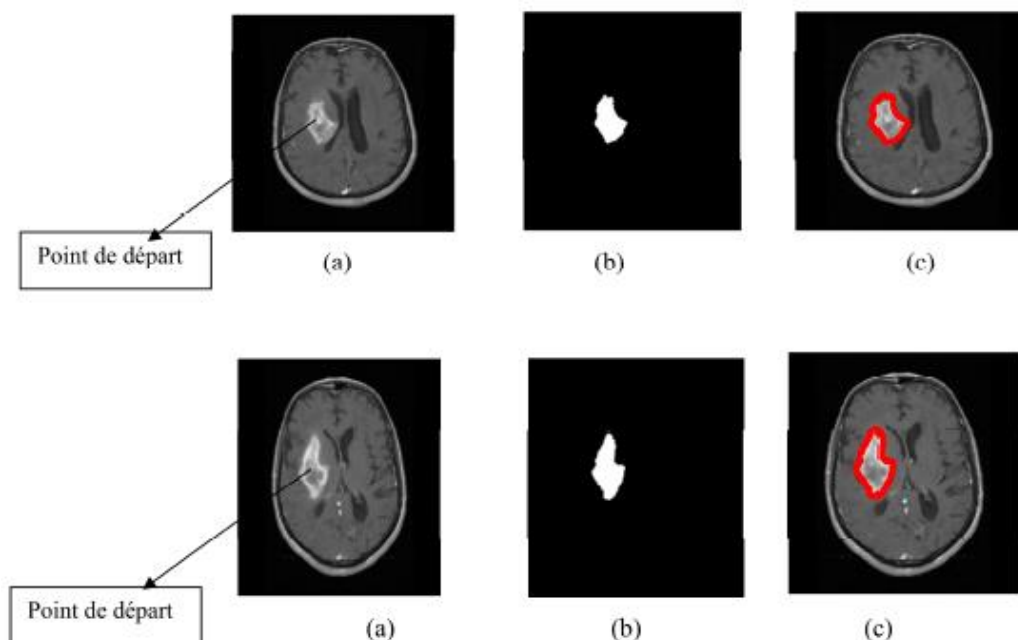


Figure II 8 Résultat de la segmentation par la méthode croissance de région, (a): image originale, (b): image binaire, (c): image segmentée.

L'approche que nous proposons se base sur la méthode de croissance de région. Elle approuvé son efficacité dans la segmentation d'image médicale. Cela s'explique par le fait qu'êtes une méthode simple à initialiser et rapide.

On remarque que cette méthode ne s'intéresse pas de lisser le contour de la région segmenter. Contrairement à d'autre méthode comme les contours actifs.

On remarque que cette méthode ne s'intéresse pas de lisser le contour de la région segmenter. Contrairement à d'autre méthode comme les contours actifs.

II.4.2.2 Segmentation par division/rassemblage (split and merge)

Le processus est décomposé en deux étapes :

a) **Division** : analyse de chaque région R_i . Si celle-ci ne vérifie pas le critère d'homogénéité, alors on divise cette région en blocs (le plus généralement en 4 quadrants) et l'on réitère le processus sur chaque sous-région prise individuellement.

b) **Rassemblage** : Si l'union de deux régions voisines (R_i, R_j) vérifie le critère d'inhomogénéité: fusion des régions.

Soit R_i ($i = 1, \dots, n$) un ensemble de régions effectuant la partition d'une image, et soit P un prédicat mesurant l'homogénéité de ces régions. Si $P(R_i) = \text{faux}$: R_i non homogène (subdivision des régions ou split).

Si $P(R_i \cup R_j) = \text{vrai}$ ($i \neq j$ et $R_i \cup R_j = \text{ensemble connexe}$): R_i et R_j sont homogènes et doivent être fusionnées (fusion des régions ou merge).

c) Application de la méthode de division fusion

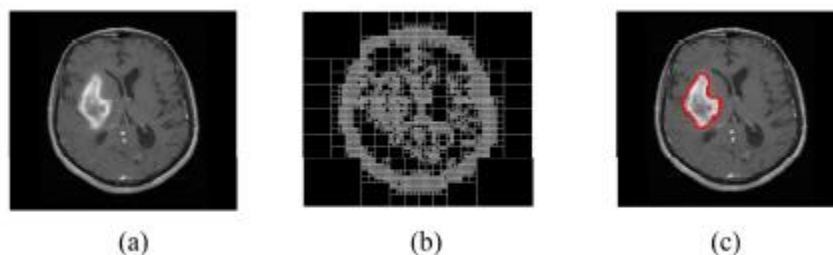


Figure II 9 Résultat de la segmentation par la méthode split and merge, (a): image initiale,

Dans cette application nous avons utilisés la méthode split and merge pour la segmentation des images cérébrales. Nous avons passés par deux étapes :

Dans la 1er partie on analyse chaque région R_i . Si celle-ci ne vérifie pas le critère d'homogénéité, alors on divise cette région en blocs (le plus généralement en 4 quadrants)et l'on réitère le processus sur chaque sous-région prise individuellement.

Dans la 2ème partie on s'intéresse à l'application de la technique de fusion : Si l'union de2 régions voisines (R_i, R_j) vérifie le critère d'homogénéité: fusion des régions.

Cette méthode "division/fusion " ne donne pas un bon résultat car nos images (images cérébrales) sont floues et nécessitent un lissage.

II.4.2.3 Segmentation par classification

Les méthodes de classification permettent de regrouper des objets en groupes ou classes d'objets plus homogènes. Les objets regroupés ont des caractéristiques communes, ils sont similaires mais se distinguent clairement des objets des autres classes [6].

Généralement les résultats des méthodes de classification sont plus ou moins différents.

II.4.2.3.1 Méthodologie de la classification

La mise en œuvre d'une procédure de classification, ayant pour objectif de classer automatiquement des objets. Cette méthode comporte généralement deux phases fondamentales :

- une phase d'apprentissage dont le but est de déterminer un espace de représentation des objets et de rechercher les paramètres discriminants capables de caractériser chaque classe d'objets ;
- une phase de reconnaissance au cours de laquelle on attribue à une classe chacun des objets inconnus dans l'espace de représentation déterminé durant l'apprentissage.

La classification peut être supervisée ou non supervisée [6].

a) Méthodes supervisées

Les méthodes de classification supervisées supposent la connaissance a priori de l'appartenance de chaque échantillon de l'ensemble d'apprentissage à une classe donnée, ce qui revient à supposer une connaissance a priori sur l'image à segmenter [7].

Dans notre cas, il s'agit la segmentation des images médicales IRM cérébrales. La classification supervisée de ces images nécessite à la création d'une base d'apprentissage pour chaque classe et pour chaque patient ce qui est en elle-même une tâche très fastidieuse pour les experts. C'est pour cette raison qu'on s'intéresse aux méthodes non supervisées.

b) Méthodes non Supervisées (Automatiques)

La méthode de classification non supervisée a pour but de trouver des partitions d'un ensemble d'individus en fonction de critères de proximité de leurs vecteurs d'attributs dans l'espace de représentation. Elles sont utilisées pour effectuer une classification en aveugle et, ainsi pour réaliser une segmentation sans connaissances a priori sur l'image. Il est par conséquent nécessaire de prendre quelques précautions. La mesure de proximité est calculée sur tous les attributs, il n'y a donc pas de réduction de l'espace de représentation. Comme il n'y a pas d'exemples d'apprentissage, si certains attributs utilisés sont non représentatifs des classes recherchées, le résultat pourra être médiocre.

II.5 Algorithme des K_means

L'algorithme k-means est l'algorithme de clustering le plus connu et le plus utilisé, d'après sa simplicité de mise en œuvre. Il partitionne les données d'une image en K clusters.

Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, k-means ne crée qu'un seul niveau de clusters.

L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des autres

des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters. Les principales étapes de l'algorithme k-means sont :

1. Choix aléatoire de la position initiale des K clusters.
2. Affecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
3. Une fois tous les objets placés, recalculer les K centroïdes.
4. Répéter les étapes 2 et 3 jusqu'à ce que plus aucune réaffectation ne soit faite

II.5 .1 Principe :

Soit un nuage de point (une image) que nous voulons partitionner en K classe. Nous posons :

x_j : le pixel j appartenant à la classe i,

y_i : le centroïde de la classe i,

L_i : le nombre de points de la classe i,

$d(x_j, y_i)$: la distance (mesure de distorsion) entre $(i) x_j$ et y_i ; dans le cas de l'algorithme K-means,

C'est la distance Euclidienne.

D_i : la distance totale de la classe i:

$$D_i = \sum_{j=1}^{L_i} d(x_j^{(i)}, y_i)$$

D : la distorsion globale pour l'ensemble des vecteurs (inertie intraclasse):

$$D = \sum_{i=1}^K D_i$$

La classification optimale est celle qui permet de minimiser la distorsion totale D. La procédure d'optimisation doit tenir compte des hypothèses suivantes :

- Pour un ensemble donné de centroïdes, la classification qui minimise distance totale D_i est celle pour laquelle chaque pixel est affecté à la classe dont le centroïde est le plus proche.
- Pour une classification donnée, il existe, pour chaque classe i, un pixel y_i qui minimise la distance totale de la classe D_i .

II.5.2 Algorithme des centres mobiles (K-means) :

1. Choisir un nombre de classes k.
2. Définir une classification C aléatoire : Choisir les k centroïdes y_i de façon aléatoire dans l'espace D des niveaux de gris $[0, L - 1]$.
3. Tant que l'inertie intraclasse n'est pas stable faire
 - 3.1. Affecter chaque niveau de gris à la classe dont le centre est le plus proche.
 - 3.2. Calculer les centres de gravité des classes de la nouvelle classification C' des classes.
 - 3.3. $C \leftarrow C'$
- Fin tant que
4. Afficher la classification obtenue.

On remarque que la tumeur apparaît bien dans ces résultats lorsqu'on utilise l'algorithme de k-means. L'utilisation plusieurs fois de l'algorithme k-means donne des résultats différents, à cause que ces attributs sont non représentatifs des classes recherchées et le résultat pourra être médiocre.

L'algorithme classique de k-means laisse un paramètre libre : le nombre de clusters, ce qui dans le cas de la segmentation d'images correspond au nombre d'intensités utilisées pour représenter l'image. Généralement le choix de k est fait empiriquement en sélectionnant la valeur de k qui minimise l'énergie. Différents critères permettent d'estimer le nombre de clusters en minimisant la distance intra-classes et à maximiser la distance interclasses (par exemple pour la segmentation d'images couleurs).

II.5 La classification floue

Le principe de la classification floue est d'affecter un élément à classer non pas à une classe comme dans les approches classiques (k-means), mais à toutes les classes avec un certain degré. Ce principe découle de ce qu'on appelle la logique floue, qui est venue compléter la logique classique [6].

II.6.1 La logique floue

Dans la vie de tous les jours, nous trouvons dans des situations où les informations dont nous disposons ne sont pas toujours aussi, des imprécisions peuvent être vues ; par exemple lorsqu'on veut établir une valeur qui caractérise le degré de similarité de deux formes.

L'être humain est habitué à utiliser des informations entachées d'incertitudes et d'imprécision dans la vie de tous les jours, il utilise ces informations incomplètes, raisonne avec elles et prend des décisions. Dans le domaine scientifique, il a été nécessaire de créer une logique qui admette des valeurs de vérité en dehors de l'ensemble {vrai,faux } pour pouvoir tenir compte et manipuler ce genre d'information incomplète.

Lukasiewicz propose en 1920 une logique ayant les trois valeurs de vérité suivantes : "Vrai", "faux" et "doute". Ces valeurs ont été représentées par l'ensemble {0,1,0.5}, étant été ensuite étendues à l'intervalle [0,1].

Néanmoins c'est Zadeh, qui à partir de l'idée d'appartenance partielle d'un élément à plusieurs classes, a introduit la logique floue [6].

II.6.2 La théorie des sous-ensembles flous

La théorie des sous-ensembles flous et les outils de raisonnement qui en découlent, proposent un cadre formel qui permet de modéliser le langage naturel et de gérer l'imprécis et l'incertain. Cette théorie est basée sur la notion de degré d'appartenance. Ces degrés sont des valeurs qui expriment l'appartenance incertaine d'un pixel à une région. Le degré d'appartenance se situe dans l'intervalle [0,1] et les classes obtenues ne sont pas forcément disjointes.

II.7 Classification floue FCM (Fuzzy C-Means)

La modélisation de l'imprécision s'effectue en considérant des frontières graduelles au lieu de frontières nettes entre les classes. L'incertitude s'exprime par le fait qu'un pixel possède aussi bien des attributs qui l'assignent à une classe qu'à une autre. La classification floue assigne donc, non pas à un pixel une étiquette relative à une classe unique, mais son degré

d'appartenance à chacune des classes. Ces valeurs expriment l'appartenance incertaine d'un pixel à une région et sont appelées degrés d'appartenance. Le degré d'appartenance se situe dans l'intervalle $[0, 1]$ et les classes obtenues ne sont pas forcément disjointes. Dans ce cas, les données X_j ne sont plus assignées à une classe unique, mais à plusieurs par l'intermédiaire de degrés d'appartenance U_{ij} du vecteur X_j à la classe i .

Le but des algorithmes de classification est non seulement de calculer les centres de classe v mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes.

Si U_{ij} est le degré d'appartenance de X_j à la classe i , la matrice $UC \times N$ est appelée matrice de C-partitions floues si et seulement si elle satisfait aux conditions :

$$\forall i \in [1, C], \forall j \in [1, N] \left\{ \begin{array}{l} u_{ij} \in [0, 1] \\ 0 < \sum_{j=1}^N u_{ij} < N \end{array} \right.$$

$$\forall i \in [1, C] \sum_{j=1}^N u_{ij} = 1$$

La fonctionnelle à minimiser J et les solutions b_i , u_{ij} , au problème du FCM sont décrites par les formules suivantes :

$$J(B, U, X) = \sum_{i=1}^C \sum_{j=1}^N (U_{ij})^m d^2(x_j, b_i)$$

$$b_i = \frac{\sum_{j=1}^N (u_{ij})^m \cdot X_j}{\sum_{j=1}^N (u_{ij})^m}$$

$$u_{ij} = \left[\sum_{k=1}^C \left(\frac{d^2(X_j, b_i)}{d^2(X_j, b_k)} \right)^{\frac{2}{(m-1)}} \right]^{-1}$$

II.7.1 Description de l'algorithme FCM

Fuzzy C-Means (FCM) est un algorithme de classification non-supervisée floue. Issu de l'algorithme des C-moyennes (C-means), il introduit la notion d'ensemble flou dans la définition des classes : chaque point dans l'ensemble des données appartient à chaque cluster avec un certain degré, et tous les clusters sont caractérisés par leur centre de gravité.

Comme les autres algorithmes de classification non supervisée, il utilise un critère de minimisation des distances intra-classe et de maximisation des distances interclasse, mais en donnant un certain degré d'appartenance à chaque classe pour chaque pixel. Cet algorithme nécessite la connaissance préalable du nombre de clusters et génère les classes par un

processus itératif en minimisant une fonction objective. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance (compris entre 0 et 1) à une classe donnée. Le cluster auquel est associé un pixel est celui dont le degré d'appartenance sera le plus élevé.

Les principales étapes de l'algorithme Fuzzy C-means sont :

1. La fixation arbitraire d'une matrice d'appartenance.
2. Le calcul des centroïdes des classes.
3. Le réajustement de la matrice d'appartenance suivant la position des centroïdes.
4. Calcul du critère de minimisation et retour à l'étape 2 s'il y a non convergence de critère [15].

II.7.2 Algorithm Fuzzy C-means (FCM)

L'algorithme des c-moyennes floues ou fuzzy c-means (FCM) repose sur la recherche des valeurs prises par les centroïdes des classes et par les degrés d'appartenance des pixels à ces classes, qui en respectant les contraintes, minimisent la fonction objectif suivante [6]:

$$J = \sum_{i=1}^c \sum_{k=1}^n U_{ik}^m d^2(x_k, v_i)$$

Avec :

- c : le nombre de classes connu à priori.
- n : la taille du vecteur de données (nombre de pixels à classer).
- u_{ik} : le degré d'appartenance du pixel x_k à la classe i connue par son centroïde v_i .
- d : le degré de similitude, il peut être la distance Euclidienne.

La méthode FCM a été largement utilisée pour la segmentation des images du cerveau. Cet algorithme utilisé pour segmenter les régions activées (tumeurs...) du cerveau, et aussi pour détecter les zones affectivement activées qui restaient silencieuses.

Dans cette application, on s'intéresse à l'extraction des différentes parties du cerveau (la tumeur, le contour du cerveau, liquide céphalo-rachidien et la partie gris et la partie blanche).

D'après les résultats obtenus on remarque que cette méthode (FCM) permet de détecter ces 4 classes à l'intermédiaire d'un degré d'appartenance et aussi grâce à la fonction de distance suivi d'un processus itératif pour minimiser cette fonction.

Cette méthode donne des meilleurs résultats pour l'extraction des différentes parties du cerveau. Comme les autres méthodes non supervisées, l'inconvénient de cet algorithme c'est comment initialiser le nombre de classes.

II.8 Comparaison entre k-means et FCM :

Nous avons testés les méthodes de segmentations Fuzzy C-means et K-means sur une série d'images en NG. D'après les résultats trouvés, k-means étant une technique d'optimisation, il s'arrête lorsqu'il y a plus de pixels à classer. Mais l'algorithme de FCM donne une représentation floue pour chaque pixel (degré d'appartenance à chaque classe).

Nous avons pu observer que, d'une manière générale, les algorithmes étudiés donnent une bonne segmentation. Néanmoins, les images comportant des défauts donnent un résultat qui semble visuellement correct. Nous avons constaté que les deux algorithmes nécessitent une

connaissance préalable du nombre de clusters. Par leur caractère itératif, ils s'avèrent inefficaces lorsque le nombre de clusters devient important.

De manière générale, nous avons pu constater que ces deux algorithmes s'avèrent efficaces lorsque les objets de l'image sont clairement séparés, mais présentant cependant deux inconvénients : d'une part, ils nécessitent le choix préalable du nombre K de clusters, ce qui rend impossible l'automatisation de la méthode ; d'autre part, ils requièrent un temps de calcul souvent élevé, du fait de leur nature itérative.

II.9 Conclusion :

Dans ce chapitre nous avons présenté les méthodes les plus utilisées pour la segmentation des images médicales. Dans notre travail, nous nous intéressons aux techniques de classification automatique (k-moyenne et Fuzzy C-means). Nous avons vu que la segmentation par classification permet de regrouper des objets en classes. Par contre, les méthodes contextuelles (croissance de régions et division–fusion) sont utilisées pour la segmentation des images, en regroupant les objets suivant les critères d'homogénéités.

Dans le dernier chapitre nous introduirons la notion de coopération en segmentation d'images.

Chapitre III :
TMS320C6416 DSK et Code
Composer Studio

III.1 Le TMS 320C6416 DSK

III.1.1 Introduction

L'objet de cette partie est d'étudier le DSP TMS320C6416. Néanmoins les informations fournies ne sont pas exhaustives et, pour une utilisation précise de telle ou telle partie, on pourra avantageusement utiliser les documentations fournies par le constructeur pour mieux détailler ce chapitre.

III.1.2 La Famille TMS 320C

Cette famille de DSP englobe des processeurs 16-32 bits, à points fixes ou flottants. Aujourd'hui, la famille TMS320 est divisée en trois Plates-formes qui sont les TMS320c2000, TMS320c5000, TMS320c6000. cette dernière englobe trois type de processeur, les TMS320c62x, TMS320c64x, TMS320c67x. On peut voir sur la figure 3.1 les différents types de processeur de cette famille (TMS320). [12]

Application	Type de DSP	Caractéristique
TMS320C6000 DSP hautes performances		
C64x	Application au traitement numérique de signal	32 bits virgule fixe 24000 MIPS - 3,0 GHz
C62x	Applications exigeantes en vitesse: stations de base des réseaux de communications mobiles, équipement de radiodiffusion, réseaux informatiques	16 bits virgule fixe architecture VLIW 1200-2400 MIPS
C67x	Applications exigeantes en précision, dynamique et vitesse: Antennes adaptatives des stations de base, imagerie médicale, reconnaissance de parole, graphisme...	32 bits virgule flottante architecture VLIW 600MFLOPS-1GFLOPS
TMS320C5000 DSP optimisés en consommation		
C54x	Applications de télécommunications exigeantes en coût, consommation, vitesse: terminaux mobile, voix sur IP,	16 bits virgule fixe 30-200 MIPS
TMS320C2000 DSP optimisé pour les applications de contrôle		
C20x	Applications de grand volume en téléphonie, électronique grand public, appareils photo numériques ou contrôleurs de disques durs...	16 bits virgule fixe mémoire flash intégrée 20-40MIPS
C24x	Applications de contrôle moteur, automatisation, robotique, contrôle d'appareils électroménagers...Bon compromis prix/performance	16 bits virgule fixe 20MIPS

Tableau III.1 : résumé les principales caractéristiques de DSP Texas Instruments

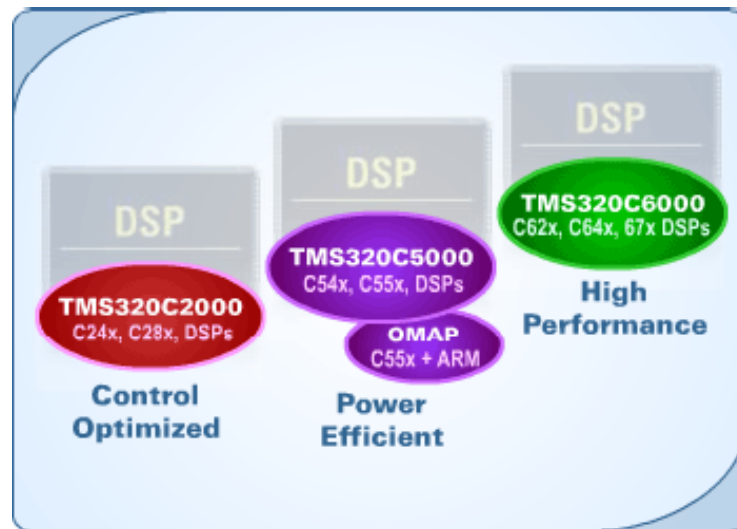


Figure III 1 La famille TMS320C

III.1.3 Les différentes familles de C6x

❖ TMS320C62X :

Ce sont les premiers C6x sortis par Texas Instruments en 1998. Avec des fréquences allant de 150 à 300 MHz, ils ont été les premiers DSP à architecture VLIW (very-long-instruction-word).

❖ TMS320C64X :

Ce sont les derniers DSP de gamme proposée par Texas Instruments. Comme les C62x, ce sont des DSP 16 bits virgule fixe. L'avantage de cette famille est qu'elle bénéficie des progrès technologiques et peut atteindre des fréquences allant de 300 à 600 MHz. Des efforts ont également été réalisés pour les transferts de données dans le processeur avec des bus 64 bits. Le jeu d'instructions a été étoffé par rapport à celui du C62x avec des instructions SIMD (single-instruction multiple-data) pour pouvoir traiter plusieurs données avec une unique instruction et ainsi diminuer la taille du programme. Ces instructions ont été spécialement ajoutées pour les applications de traitement d'images et pour l'utilisation du bus interne 64 bits. [13]

❖ TMS320C67X :

Ce sont les DSP à virgule flottante de la gamme. Les données sont de 64 bits, et les bus internes également. Les fréquences proposées vont de 100 à 225 MHz. Le jeu d'instructions est identique à celui du C62x, auquel ont été ajoutées des instructions pour les calculs spécifiques en virgules flottantes. Actuellement, aucune nouvelle version n'est annoncée. [4]

III.1.4 Présentation de la plateforme TMS 320C6000

Avec une performance jusqu'à 4800 millions instructions par second (mips) et un compilateur C effectif, les TMS320C6000 DSP donnent à l'architecte du système, une possibilité illimitée

de différencier son produit d'un autre. La Haute performance, et les prix accessibles, font de la plateforme TMS320C6000 l'idéale solution pour multicanal, demande multifonction tel que:

- Stations de base de la ligne locale sans fil.
- Serveurs à accès à distance (RAS).
- Ligne d'abonnement Numérique (DSL) système.
- Modems câblés.
- Systèmes de la téléphonie multivoies.

La plateforme TMS320C6000 est aussi une solution idéale pour exciter des nouvelles applications, par exemple:

- Protection de maison personnalisée avec type de caractère et reconnaissance de main "empreinte digitale".
- Contrôle de la croisière avancée avec navigation GPS et évation de l'accident.
- Diagnostics médicaux éloignés.
- Faisceau-formant stations de base.
- Reconnaissance vocale.
- Audio.
- Radar.
- Modelage atmosphérique.

III.1.5 Caractéristiques principales du TMS 320C6416

Le C6416 DSK est une plateforme du développement autonome bas-prix qui permet aux utilisateurs, d'évaluer et développer des applications pour le TI C64xx DSP famille

Le DSK sert aussi comme hardware référence du matériel pour le TMS320C6416 DSP

Dans notre cas, le DSP est monté sur une carte Starter Kit de Texas Instrument munie des synchrones :

- Le TMS320C6416 DSP fonctionnant à 600 MHz.
- Un convertisseur audio bidirectionnel de type « CODEC » AIC23 stéréo.
- 16 M octets de DRAM synchrones.
- 512 K octets de mémoire flash non volatile.
- 4 LEDs et des switches DIP.
- Le software embarqué d'auto-configuration de la carte sur un asic programmable de type CPLD.
- Options de démarrage configurables
- Connecteurs d'extension standard pour carte fille
- un émulateur JTAG embarqué.
- une interface USB avec l'ordinateur hôte.
- Alimentation monophasée (+ 5V)

III.1.6 Architecture générale du starter KIT TMS 320C6416

III.1.6.1 Le Processeur :

Les DSP TMS320C64x sont les DSP à point fixe de la plus haute performance dans la plateforme DSP TMS320C6000™. Le TMS320C64x™ est basé sur l'architecture de haute performance de deuxième génération (VLIW) (très-long-instruction-mot) développée par Texas Instruments. Le C64x™ est un Compatible avec le code de la plate-forme C6000™ DSP.

C64x Offrent des solutions rentables aux défis de la programmation DSP hautes performances jusqu'à 5760 millions d'instructions par seconde (MIPS) à une fréquence d'horloge de 720 MHz, le. Les DSP C64x Possèdent la flexibilité opérationnelle des contrôleurs à grande vitesse et la capacité numérique des processeurs de réseau.

Le processeur de cœur C64x™ DSP possède 64 registres à usage général de longueur de mot de 32 bits et huit unités fonctionnelles indépendantes deux multiplicateurs pour un résultat 32 bits et six unités logiques arithmétiques (ALU) avec Extensions VelociTI.2™. Les extensions VelociTI.2™ des huit unités fonctionnelles comprennent de nouvelles d'accélérer les performances dans les applications clés et d'étendre le parallélisme de l'architecture VelociTI™. Le C64x peut produire quatre multi-accumulations (MAC) de 16 bits par cycle pour un total de 2880 millions de MAC par Seconde (MMACS) ou huit MACs 8 bits par cycle pour un total de 5760 MMACS.*

Le DSP C64x a également la logique matérielle spécifique à l'application, la mémoire sur puce et d'autres périphériques sur puce semblables aux autres C6000™ DSP

Le C6416 dispose de deux coprocesseurs intégrés haute performance [ViterbiDecoderCoprocesseur (VCP) et le coprocesseur de décodage de Turbo (TCP)] qui accélèrent considérablement des opérations de décodage de canal sur la puce. Le VCP fonctionnant à l'horloge CPU divisée par 4 peut décoder plus de 600 7.95-Kbps adaptative multi-taux (AMR) [K = 9, R = 1/3] canaux vocaux. Le VCP supporte les longueurs de contrainte K = 5, 6, 7, 8 et 9, les vitesses R = 1/2, 1/3 et 1/4, Et polynômes flexibles, tout en générant des décisions difficiles ou des décisions souples. Le TCP fonctionnant à l'horloge CPU Divisé par 2 peut décoder jusqu'à quarante-trois canaux 384-Kbps ou sept canaux codés turbo de 2 Mbps (en supposant que 6 Itérations). Le TCP implémente l'algorithme max * log-map et est conçu pour supporter tous les polynômes et Les tarifs exigés par les projets de partenariat de troisième génération (3GPP et 3GPP2), avec cadre entièrement programmable Longueur et turbo entrelaceur. Les paramètres de décodage tels que le nombre d'itérations et les critères d'arrêt sont Également programmable. Les communications entre le VCP / TCP et la CPU s'effectuent via l'EDMA manette. [14]

functional block and CPU (DSP core) diagram

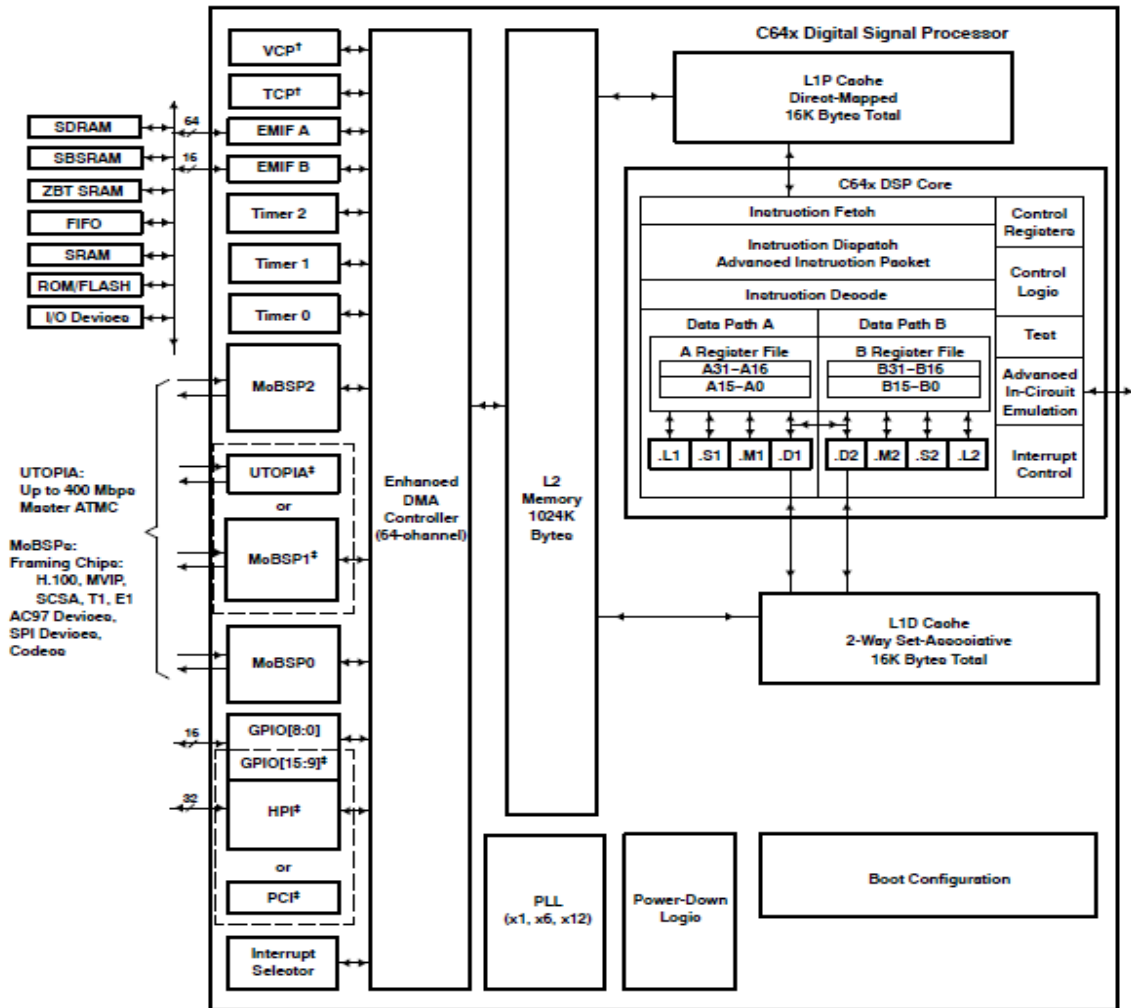


Figure III 2 Architecture de TMS320C6416 DSP

Le C6416 emploie une architecture cache-basée à deux niveaux, et a un ensemble puissant de divers périphériques. La cache de programme du niveau 1 (L1P) est une cache ‘dirigé-tracé’ par 4K-Byte, et la cache de données du niveau 1 (L1D) est une cache 4K-Byte placer-associative bidirectionnelle. La mémoire du niveau 2 ou la cache (L2) se compose d'un espace mémoire 256K-Byte, partagé entre le programme et l'espace de données. Les 64K bytes des 256K bytes dans la mémoire L2 peuvent être configurés en tant que mémoire, cache, ou combinaison tracée des deux. Les bytes 192K restant dans L2 sert de SRAM tracé. Le C6416 possède un périphérique riche qui inclut :

- ✓ Deux (McASP) ‘portes série audio multicanales ‘ ‘multichannel audio serial port’.
- ✓ Deux portes série protégées multicanales (McBSP) ‘ ‘multichannelbuffered serial port’.
- ✓ deux ‘Inter-Intégrés - circuit’les bus (I2C),
- ✓ un module à usage universel consacré d'entrée-sortie (GPIO) ‘ ‘ General-Purpose-Input/Output ‘.

- ✓ Deux temporisateurs à usage universel, une interface de centre-port (HPI) ‘‘ Host Port Interface ‘‘.
- ✓ Une interface externe glueless mémoire (EMIF) ‘‘External Memory Interface ‘‘ incluant SDRAM, SBSRAM, et périphériques asynchrones. La porte série soutient le multiplexage temporel sur chaque goupille de 2 à 32 fentes de temps.

Le C6416 a une largeur de bande suffisante pour soutenir chacune des 16 goupilles périodiques de données transmettant un signal stéréo de 192 kilohertz. Des données périodiques dans chaque zone peuvent être transmises et reçues sur les goupilles périodiques multiples de données simultanément et être composées dans une multitude de variations sur le format du bruit d'Inter-IC de Philips (I2S). En outre, l'émetteur de McASP peut être programmé pour produire S/PDIF multiple, IEC60958, AES-3, et les canaux de données codés par CP-430 simultanément, avec une RAM simple contenant la pleine exécution des champs de statut de données et de canal d'utilisateur.

Le McASP fournit également les dispositifs étendus de contrôle d'erreurs et de rétablissement, tels que le mauvais circuit de détection d'horloge pour chaque horloge principale à haute fréquence et qui vérifie que l'horloge principale est dans une marge de fréquence programmée.

Les deux ports d'I2C sur le TMS320C6416 permettent au DSP de commander facilement les périphériques et de communiquer avec un processeur du centre serveur. En outre, la porte série protégée multicanal standard (McBSP) peut être employée pour communiquer avec les périphériques de mode périphérique périodique de l'interface (SPI). Le dispositif TMS320C6416 a deux bootmodes, du HPI ou de la ROM asynchrone externe.

La génération de TMS320C64x DSP est soutenue par l'ensemble d'eXpressDSP™ de TI, outil de développement de repère d'industrie, y compris un compilateur fort de linéarisation de C/C++. L'environnement de développement intégré par ‘‘Studio™ de compositeur de code’’ (IDE), ‘‘Studio Code Composer’’ (CCS), émulateur JTAG-basée et correction en temps réel, et le grain de DSP/BIOS™.

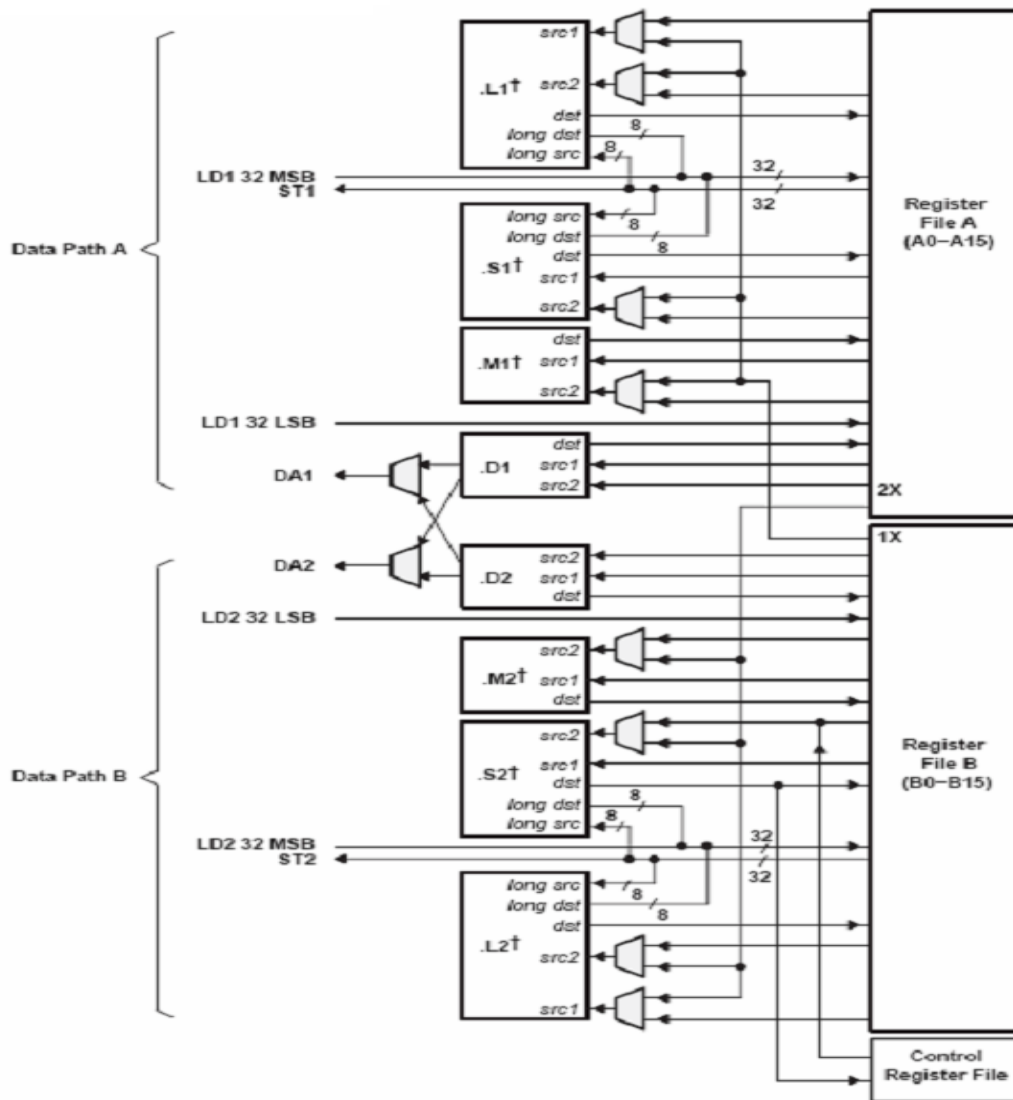


Figure III 3 TMS320C6713 CPU Data path

La figure montre les deux « data paths » (voies accès de donnée) du TMS320c67 et apparaissent :

- Les deux file registre (A et B).
- Les huit unités (L1, L2, M1, M2, S1, S2, D1, et D2).
- Les deux voies de chargement depuis la mémoire LD1 et LD2 (LD=load).
- Les deux voies de stockage vers la mémoire ST1 et ST2 (ST=store).
- Deux croisements (registre file cross paths) entre les files de registre 1X et 2X.
- Deux chemins d'adresse de données (data adresse paths) DA1 et DA2.

III.1.6.2 Starter KIT TMS320C6416 :

Le C6416 DSK est une plateforme du développement autonome bas-prix qui permet aux utilisateurs, d'évaluer et développer des applications pour le TI C67xx DSP famille. Le DSK sert aussi comme hardware référence du matériel pour le TMS320C6416 DSP.

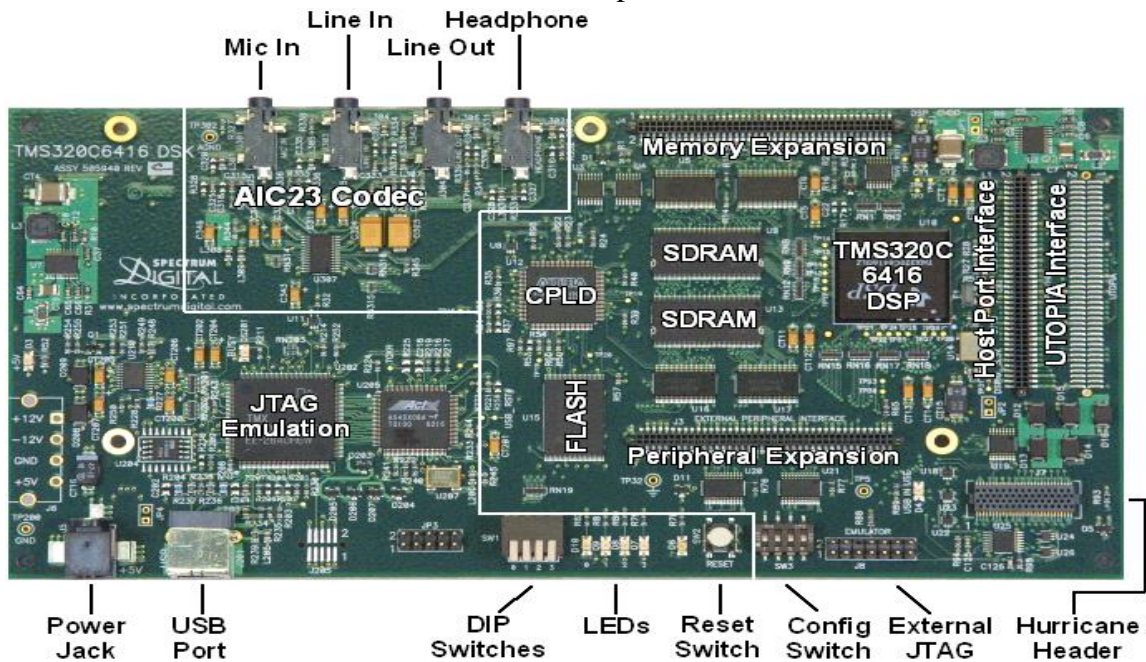


Figure III 4 La Carte Starter Kit (C6416DSK)

La figure 3.4 représente le corps du microcontrôleur intégrant la CPU, les périphériques et les interfaces externes :

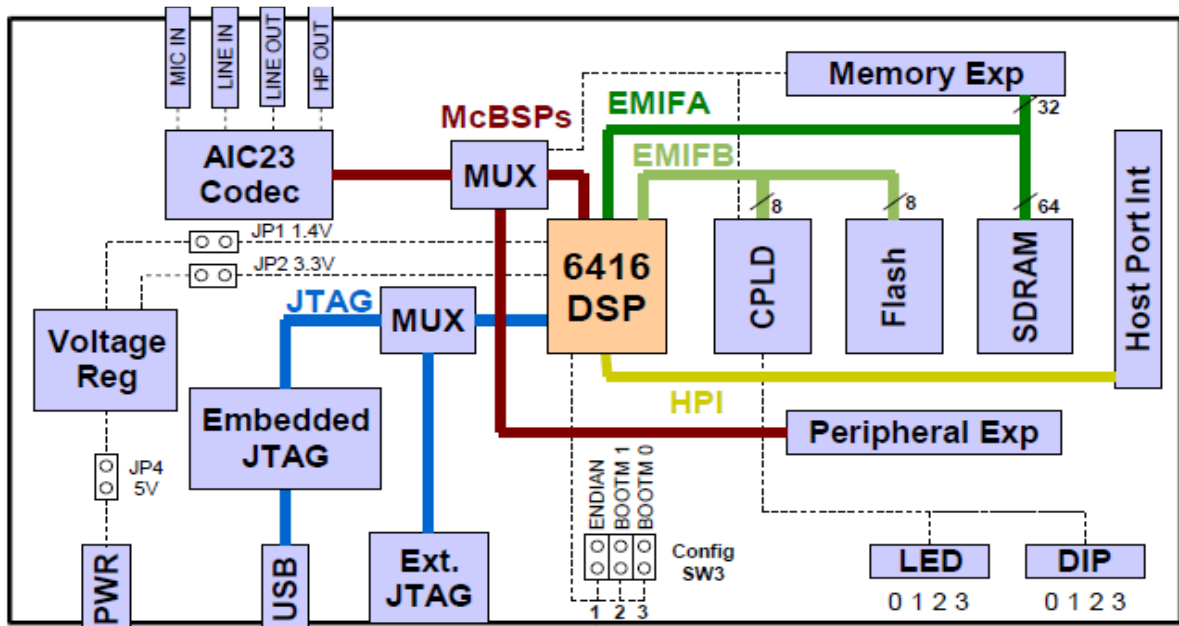


Figure III 5 Schéma globale de la carte C6416 DSK

III.1.7 Fonctionnement :

Le DSP sur les 6416 interfaces de DSK aux périphériques à bord par un EMIF de 32 bits. Le SDRAM, le flash et le CPLD sont reliés au bus.

Les signaux d'EMIF sont des connecteurs également branchés d'expansion de daughtercard 'carte fille ' qui sont utilisés pour les panneaux addition de tiers. Le DSP connecte aux signaux audio analogues par un codec AIC23 à bord et quatre crics audio de 3.5 millimètres (entrée de microphone, ligne entrée, ligne de sortie, et la sortie d'écouteur). Le codec peut choisir le microphone ou la ligne entrée comme entrée active. La sortie analogique est conduite à la ligne sortie (gain fixe) et aux connecteurs d'écouteur (gain réglable).

Un programmable logique dispositif appelé un CPLD est utilisée pour mettre en application la 'glue logic' qui attache les composants entre eux. Le DSK inclut 4 LED et 4 positions DIP interrupteur comme manière simple de fournir à l'utilisateur la rétroaction interactive. Tous les deux sont consultés par la lecture et l'inscription aux registres de CPLD.

Une alimentation de l'énergie 5V externe incluse est employée pour actionner le panneau. Les régulateurs de tension à bord de commutation fournissent la tension de noyau de +1.26V DSP et des approvisionnements de +3.3V I/O. Le panneau est tenu dans la remise jusqu'à ce que ces approvisionnements soient selon des caractéristiques de fonctionnement.

Le code composer communique avec le DSK par un émulateur JTAG incorporé avec une hôte interface d'USB. Le DSK peut également être utilisé avec un émulateur externe par le connecteur externe de JTAG. [14]

III 2 Code Composer Studio

III 2 .1. Introduction

La carte DSK est supportée par le logiciel Code Composer Studio qui contient des outils de développement, y compris un compilateur de haute optimisation de C/C++, émulation basée sur le JTAG, débogueur à temps réel (real-time debugging).

III 2.2. Présentation du Code Composer Studio CCS

Le Studio Code composer est un compilateur conçu exclusivement pour les C6x et C5x, développé par Texas Instrument Il s'agit :

- **Environnement de développement intégré (IDE)** : il permet l'édition ('built'), et la correction ('debug') des programmes destinés au DSP.
- **Outils de génération du code pour le TMS320C6000** : ces outils sont le compilateur, l'assembleur et l'éditeur de lien.

- **Le compilateur C/C++** permet de compiler le programme source (xxx.c) pour le convertir en assembleur (xxx.asm), l'assembleur reçoit le fichier xxx.asm et le convertit en langage machine ou fichier objet (xxx.obj), enfin l'éditeur de liens (linker) qui combine les fichiers objet et les fichiers bibliothèques et le fichier xxx.cmd pour produire un fichier exécutable avec une extension .out, c'est ce fichier qui sera chargé sur le processeur C6416 pour être exécuté.

- **DSP/BIOS**: c'est un outil d'analyse en temps réel, pour s'en servir, on doit créer un fichier de configuration 'xxx.cdb', où seront définis les objets utilisés par l'outil DSP/BIOS. Ce fichier permet aussi de faciliter l'organisation de la mémoire et la gestion du vecteur des interruptions, en offrant la possibilité de les faire sur un environnement visuel via la section de gestion de la mémoire MEM (Memory Section Manager), et via HWI (Hardware Interrupt Service Routine Manager) pour les interruptions.

- **JTAG** (Joint Team Action Group) et RTDX (Real Time Data Exchange) le RTDX permet un échange de données en temps réel entre l'hôte (PC par exemple) et la destination (la carte DSK dans notre cas), il permet aussi l'analyse et la visualisation des données au cours de l'exécution du programme, alors que le lien JTAG est utilisé pour atteindre l'émulateur (qui se trouve à l'intérieur du DSP), c'est ce dernier qui permet au CCS de contrôler en temps réel l'exécution du programme.

- **Simulateur intégré** : le logiciel Code Composer Studio offre la possibilité de tester des programmes pour DSP sans utiliser la carte à l'aide d'un simulateur intégré. L'application CCS fournit tous les outils logiciels nécessaires pour le développement du DSP.

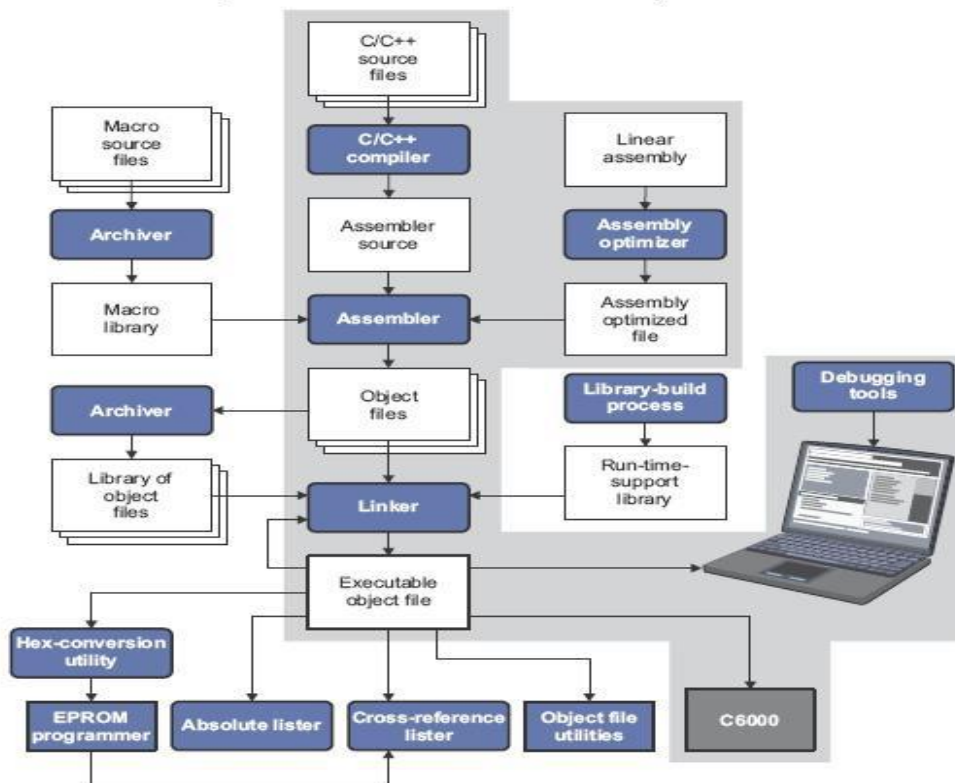


Figure III 6 Organigramme d'un système de développement de logiciel pour DSP

III 2 .3. Configuration du développement et environnement

Les exécutables sur DSP sont élaborés en utilisant des environnements de développement intégrés (IDE) fournies par les fabricants des DSP, ils intègrent de nombreuses fonctions, telles que l'éditeur, le débogage, la gestion des fichiers de projet, et le profilage. L'environnement de développement pour Texas Instrument (TI) est appelé « Code Composer Studio ». le constructeur TI met à disposition gratuitement le compilateur, l'assembleur et l'éditeur de liens optimiseur pour une utilisation non commerciale.

La figure 4.2 montre un exemple d'un écran typique du Code Composer. Sur le côté gauche il y a la liste de tous les fichiers inclus dans le projet. Au centre de l'écran deux fenêtres affichent le code, comme un fichier C (process.c) et le code assembleur (fenêtre Disassembly). L'exécution s'est arrêtée sur un point d'arrêt (breakpoint) activé au préalable. A dessous des fenêtres de codes, deux fenêtres de mémoire sont également visibles, détaillant les données présentes à partir des adresses 0x80000000, et à partir des adresses 0x40000030. La donnée à l'adresse 0x80000002 est d'une couleur différente parce que sa valeur a changé récemment. Au bas de l'écran de l'IDE, on trouve :

- la fenêtre de compilation/Link, qui donne les résultats de la dernière compilation du code ;
- la fenêtre Watch, qui affiche les valeurs prise par deux variables
- et la fenêtre Register, qui affiche le contenu de tous les registres du DSP.

Sur le côté droit il y a trois graphiques: les jaunes montrent les régions de mémoire, tandis que le vert représente la transformée de Fourier rapide de données stockées en mémoire, tel que calculé par l'IDE. [13]

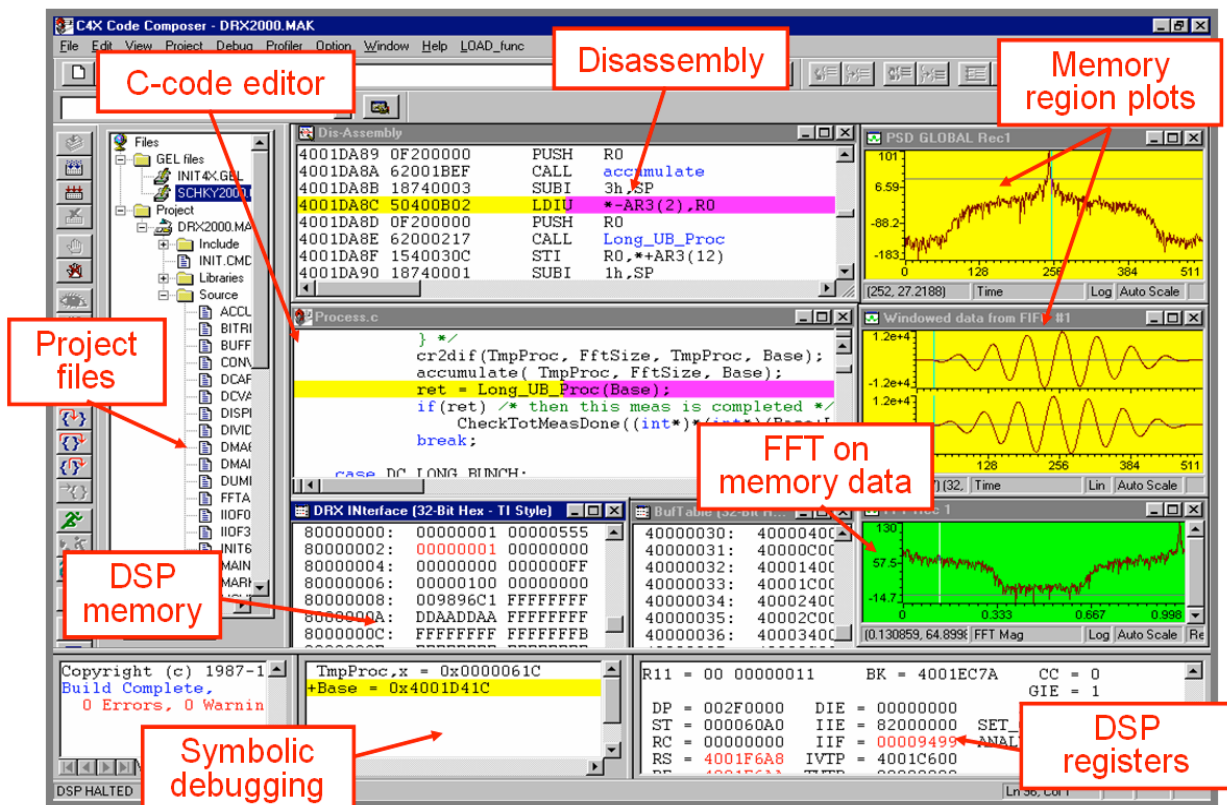


Figure III 7 Capture d'écran du Code Composer Studio

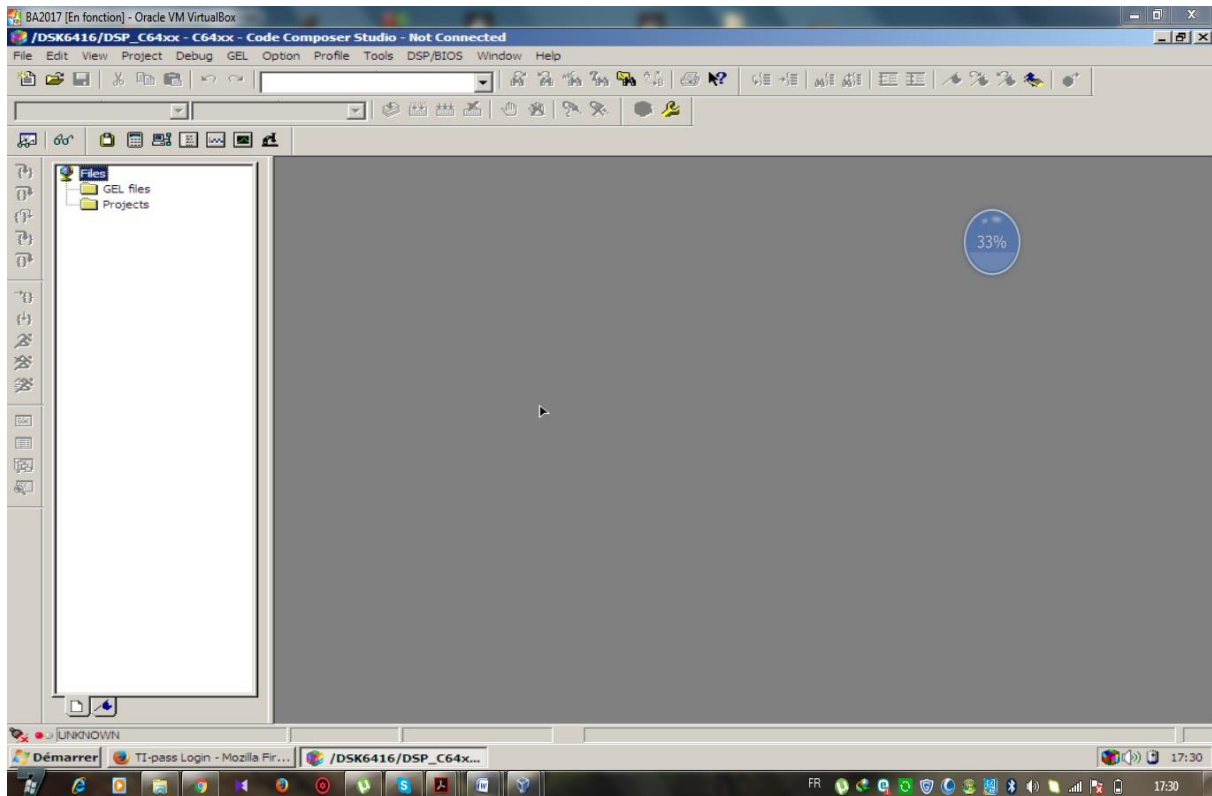


Figure III 8 Code composer studio

III 2.4. Création du fichier exécutable

Pour créer le fichier exécutable qui sera charge sur le DSP, il faut passer par plusieurs étapes :

1. Dans le menu principal, on clique sur « Project », et on choisit « New ». Dans cette fenêtre, on donne un nom pour notre projet,

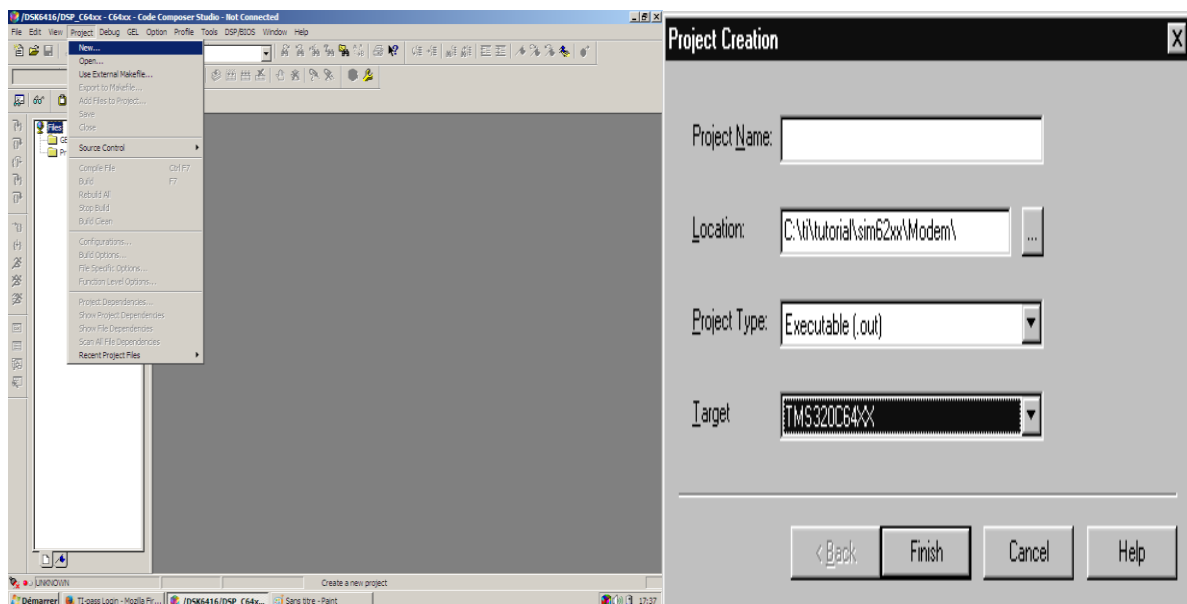


Figure III 9 Création du fichier

2. Il faut écrire le programme principal en C ou en Assembleur en cliquant dans le menu principal sur « File » puis « new ».

3. Il faut joindre tous les fichiers supports pour le bon fonctionnement du programme, pour le faire, on doit cliquer dans le menu principal sur « Project », puis « add files to project », et finalement sur « open ».

4. Joindre aussi le fichier de commande (.cmd ou .lcf). Ce type de fichier divise la mémoire de la carte en des sections.

5. Avant de générer le code, il faut configurer les options de compilation et de lien

(Project → build options).

6. Après l'étape de configuration, on construit le projet (Project → Rebuild All) qui génère le fichier exécutable avec l'extension .out. Ce dernier est ensuite chargé sur le DSP (File → Load) et finalement, il ne reste qu'à exécuter le programme

(DebugRun).

III 2 .5. Les Fichiers supports

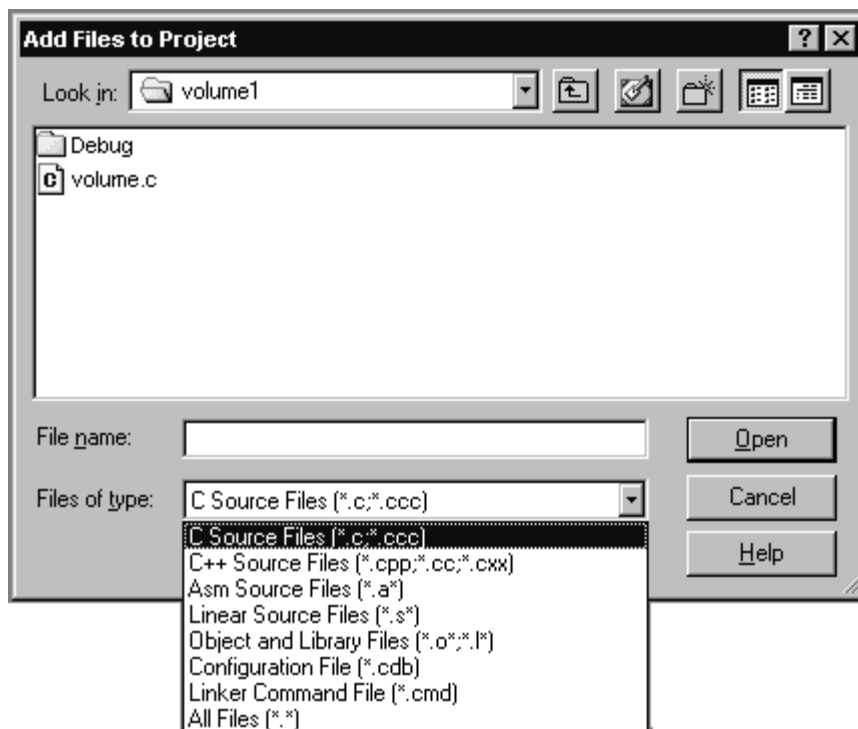


Figure III 10 Les Fichiers supports

1. **C6416 dskinit.c**: contient les fonctions initialisant le DSK, le codec, les ports séries et les entrées/sorties. Il n'est pas inclus avec CCS.

2. **C6416dskinit.h**: fichier en-tête avec des fonctions prototypes. Des fonctionnalités telles que celles utilisées pour sélectionner l'entrée micro au lieu d'entrée- ligne (par défaut), le gain d'entrée, et ainsi de suite, elles sont toutes obtenues à partir de ce fichier.
3. **C6416dsk.cmd**: fichier de commande. Ce fichier peut être modifié lors de l'utilisation de la mémoire externe à la place de la mémoire interne.
4. **Vectors_intr.asm**: une version modifiée d'un « vector file » inclus avec le CCS pour gérer les interruptions. Douze interruptions, INT4 à INT15, sont disponibles, INT11 est sélectionnée pour ce « vector file ». Elles sont utilisées pour les programmes d'interruption pilotés.
5. **Vectors_poll.asm**: c'est un « vector file » pour les programmes utilisant le polling.
6. **rts6700.lib**, **dsk6713bsl.lib**, **cs16416.lib**: run-time, la carte, et bibliothèques supports pour la puce, respectivement. Ces fichiers sont inclus avec le CCS.

Ces trois fichiers d'extension .lib sont localisés dans : C6000\cgtools\lib, C6000\dsk6416\lib, et c6000\bios\lib, respectivement.



Figure III 11 Les Fichiers supports dans le code composer studio

III 2 .6. Programmation à l'aide du Matlab/Simulink

Simulink est un logiciel de simulation de systèmes dynamiques, il s'agit d'une extension du logiciel de calcul Matlab qui propose de nombreuses bibliothèques de model intéressant. Le tableau suivant indique les versions de Matlab et de code composer studio qui sont compatibles entre eux :

Version Link for CCS	Version Matlab	Version CCS
3.0	2007A	<ul style="list-style-type: none"> • CCS 3.2 pour C64xx • CCS 3.1 pour C2000 ; C5000 ; C6000
2.1	R2006B	<ul style="list-style-type: none"> • CCS 3.2 pour C64xx • CCS 3.1 pour C2000 ; C5000 ; C6000
2.0	R2006A+	<ul style="list-style-type: none"> • CCS 3.1 pour C2000 ; C5000 ; C6000
1.5	R2006A	<ul style="list-style-type: none"> • CCS 3.1 pour C2000 ; C5000 ; C6000
1.4.2	R14SP3	<ul style="list-style-type: none"> • CCS 3.0 pour C6000 • CCS 2.2 pour C2000 ; C5000 ; C6000

Tableau III.2 : Compatibilité entre Matlab et CCS

En utilisant Real-Time Workshop, il sera permis de générer un code efficace pour les processeurs C6000 directement à partir des modèles Simulink. L'embedded Target for TI C6000 DSP permet le prototypage rapide des logiciels temps réel pour DSP Texas instruments (TI) C67xx a virgule flottante et C62x et C64x a virgule fixe.

III 2 .7. Real-Time Workshop

Real-Time Workshop est un outil de Matlab qui permet de générer automatiquement un code 'C' à partir d'un modèle simulink. De plus, il permet de produire un code optimisé, portable et personnalisable d'après les modèles générés par simulink. Il supporte tout type de systèmes, c'est-à-dire continus, discrets ou hybrides. Il permet la simulation du système étudié en mode externe, c'est-à-dire en s'interfaçant avec un pc cible.

III 2 .8. Conclusion

La programmation des DSP nécessite de vastes connaissances techniques (électronique, informatique, etc.), et il en résulte de nombreuses applications dans des domaines les plus Variés, l'importance du langage Cd ans les systèmes DSP a considérablement augmenté au cours des derniers ans. En outre, T Ia travaillé dur pour fournir facile à utiliser, Bien qu'étant le plus répandu, le langage C n'est pas le seul utilisable pour programmer un DSP, il existe quelques rares compilateurs ADA, FORTRAN et PASCAL.

Dans la deuxième partie de ce chapitre nous avons présenté le logiciel « code composer studio » et comment implémenter des filtres numériques sous DSP plate-forme

Chapitre VI :
Implémentation d'algorithme
développé sur TMS 320C6416
DSP

IV.1 Introduction

Dans ce chapitre, nous nous intéressons à la réalisation d'une technique de segmentation des images par résonance magnétique (IRM). Ce chapitre a pour objectif de développer une méthode de segmentation afin de rendre cette dernière plus rapide et plus efficace pour extraire la partie tumorale du cerveau. Nous détaillons dans ce chapitre les différentes étapes de cette approche, pour effectuer ce processus. On a développé une fonction MATLAB puis nous avons créé un modèle Simulink pour passer à l'implémentation sur TMS320C6416 DSK.

IV.2 Présentation du Model Simulink Proposé pour l'implémentation

VI.2.1 Présentation du Model SIMULINK

Le model présente une architecture efficace pour la segmentation d'une image. l'architecture proposée offre une alternative a travers un outil graphique MATLAB. l'objectif principale de l'utilisation d'un outil avec une interface graphique de haute niveau sous les blocks en fonction MATLAB SIMULINK qui le rend très facile a manipuler par rapport au d'autre , il existe des nombreuse différentes technique pour la segmentation d'image ,mais tout ce que les technique exigent un langage de haut niveau , on peut construire notre model propose pour la segmentation d'image . ce model utiliser des différent block de " video and image processing blockset", on présent ces différent blocks comme suit :

Blocks	Library	Quantity
Image From File	Video and Image Processing Blockset> Sources	01
Video Viewer	Video and Image Processing Blockset> Sinks	02
Embedded MATLAB Function	Simulink> User-Defined Functions	04
Image data type conversion	Video and Image Processing Blockset> Conversions	02
Color space conversion	Video and Image Processing Blockset> Conversions	01

TABLE IV.1 Différents blocs Simulink Utilisés

VI.2.2 architecture de model simulink

L'architecture de notre Model Simulink proposé est présentée comme suit

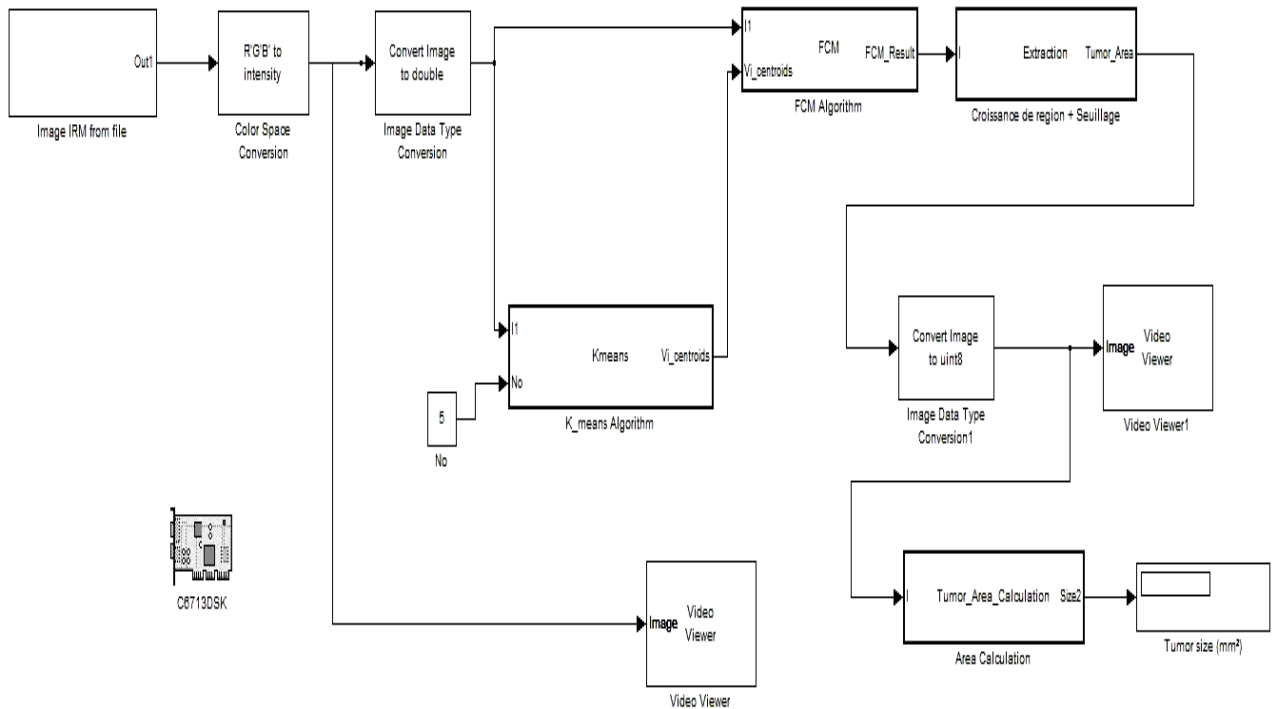


Figure IV 1 Architecture du Model Simulink proposé

IV. 3 Implémentation des Algorithmes sur TMS320C6713 DSK

Notre modèle simulink développé utilise des différents Blocksets, on présente ces différents blocks comme suit :

a- Target préférences /C6416DSK

La génération DSP à point fixe à haut rendement dans la plate-forme DSP TMS320C6000™ développé par Texas Instruments ce qui rend ces DSP un excellent choix pour les applications multicanaux et multifonctions. MATLAB a développé un outil "Link for code composer studio" qui lui permet de communiquer avec le code composer studio et on utilisant cet outil ; le model SIMULINK est traduit en projet écrit en C qui pourra être exécuté sur CCS. Sur SIMULINK il y a toute une librairie contenant des blocks spécifiques pour chaque type de carte DSP, dans notre model on ajoute le block C6416, et on choisit le type de la carte et le type de processeur et la fréquence d'horloge

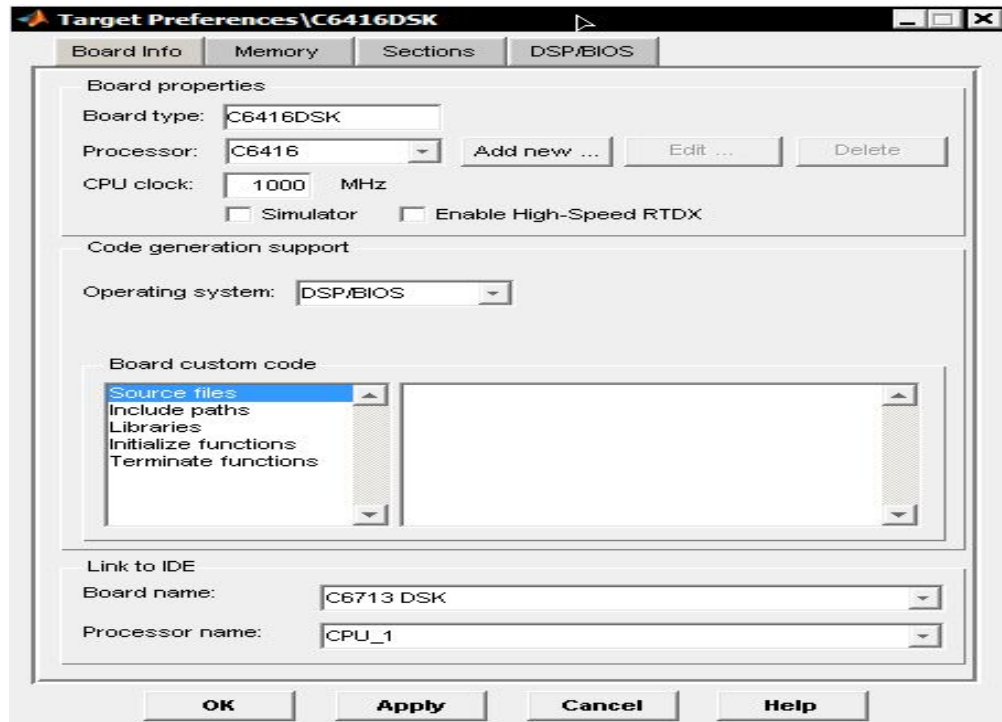


Figure IV 2 Target préférences /C6416DSK

b- Colorspace conversion

- Nous pouvons utiliser ce block pour convertir toute image R'G'B 'en une image d'intensité

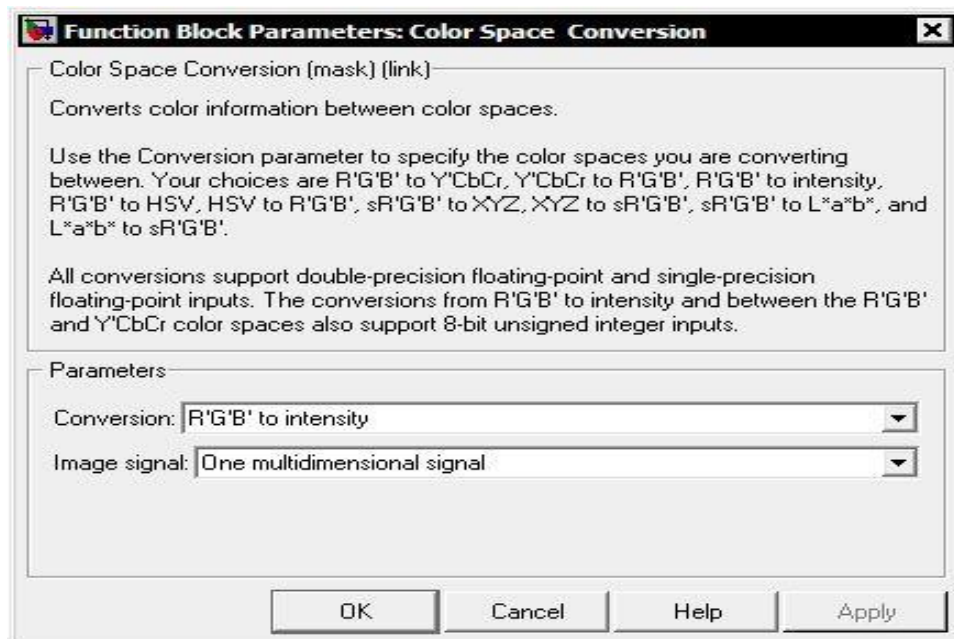


Figure IV 3 Colorspace conversion

c- Image data type conversion

Convertir l'image en double précision, Don Notre case l'image se codé sur 8bits elle deviendra sur 16 bits

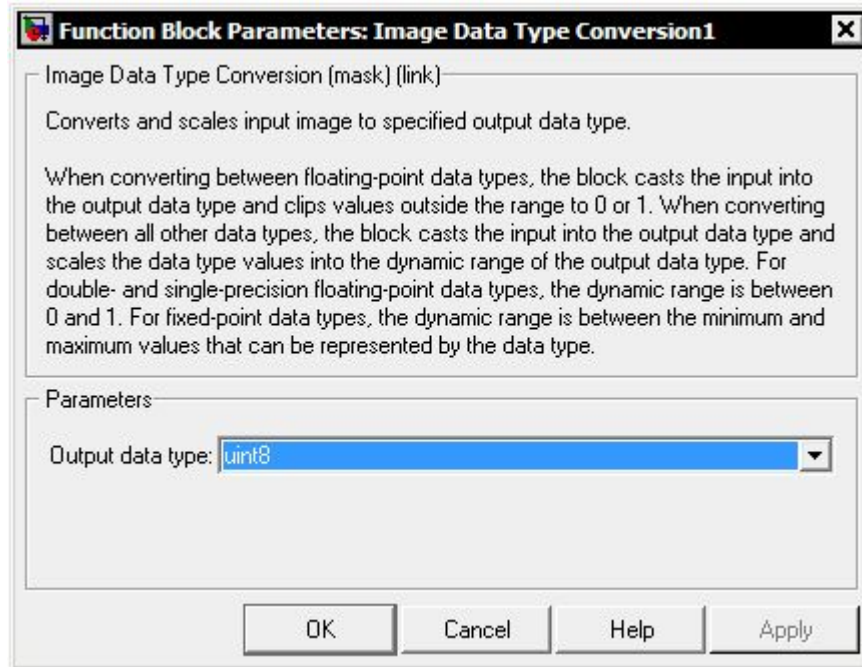


Figure IV 4 Image data type conversion

d- Image data type conversion

Convertir l'image en double précision, dans notre cas l'image e codé sur 8bits elle deviendra sur 16 bits

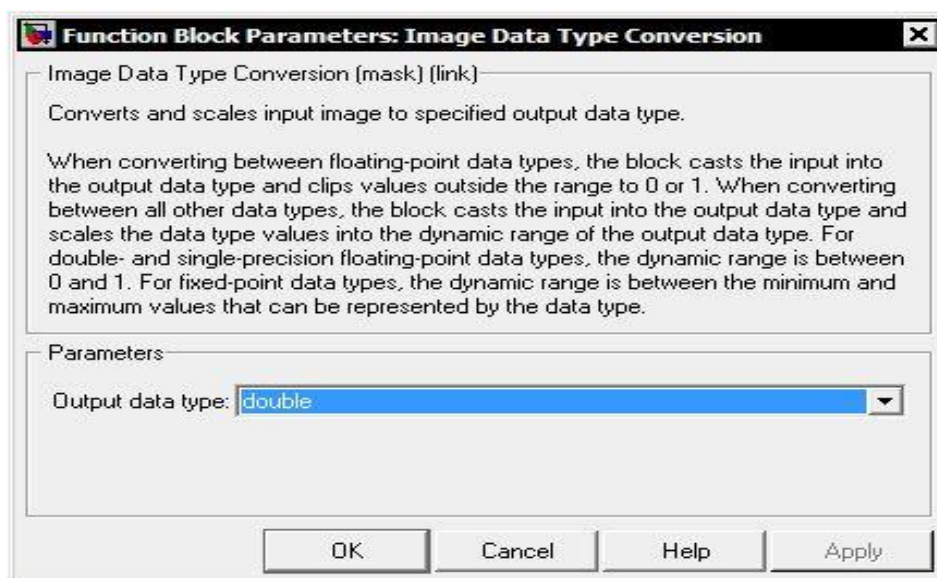


Figure IV 5 Image data type conversion

e- Image data type conversion 1

C'est aussi un block de conversion mais cette fois il fait l'inverse du premier pour qu'on puisse visualiser l'image avec le block videoviewer

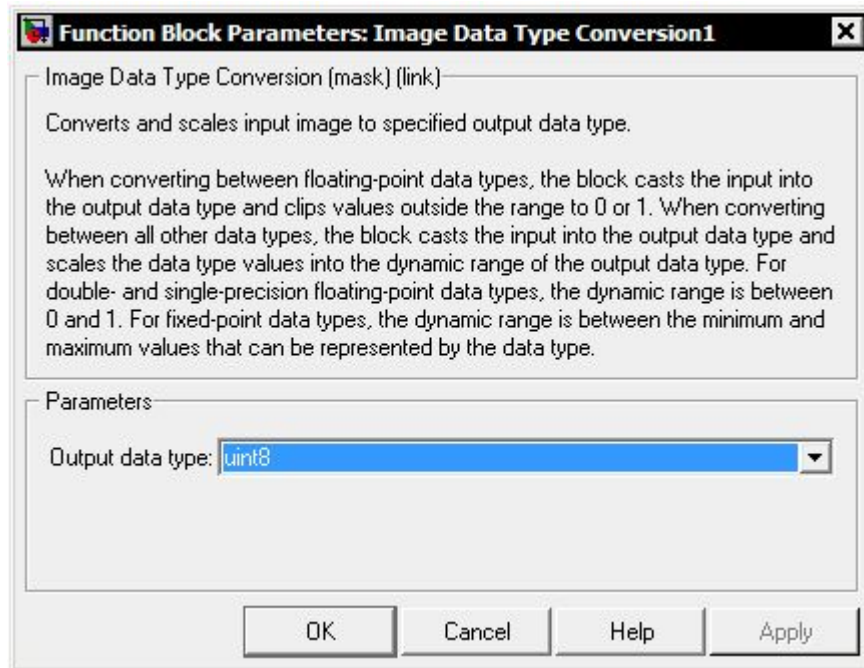


Figure IV 6 Image data type conversion 1

f- Vidéo Viewer

En utilisant Video Viewer, Nous pouvons sélectionner l'image que nous souhaitons lire ou bien visualiser.

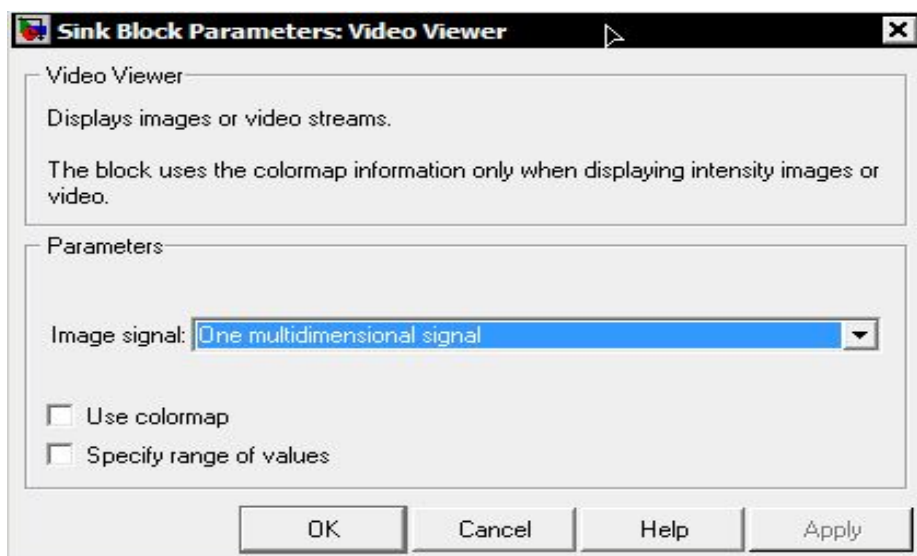


Figure IV 7 Vidéo Viewer

g- Sink block parameters :tumor size

Ce bloc nous permettre de définir la taille exacte de la tumeur cérébrale

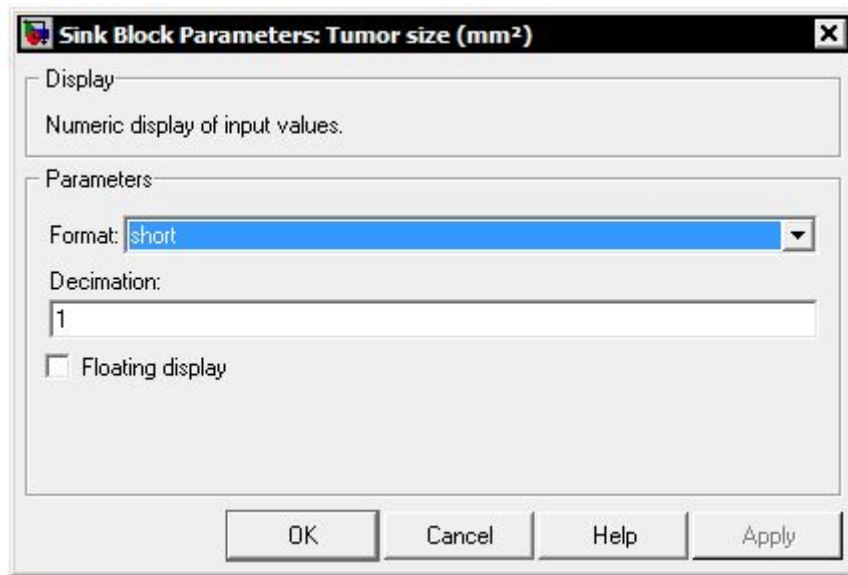


Figure IV 8 Sink block parameters :tumor size

IV.4 Simulation et résultats obtenus

IV.4.1 Instruction pour la connexion de la carte

Il faut suivre les étapes et les instruction indiquée ci-dessous pour ne pas avoir des problème au cours de déroulement de l'implémentation.

- Assurer que le code composer studio CSS est fermé.
- Brancher l'alimentation de la carte DSK : d'abord l'alimentation au secteur, puis l'alimentation continue à la carte.
- Attendre que les LEDs aient clignoté puis restent allumées.
- Brancher le câble USB.
- Lancer CCS (Code Composer Studio)
- Se connecter à la cible : Debug > Connect

Une icône verte doit apparaître en bas à gauche de la fenêtre de CCS pour préciser que la carte est bien connectée au PC.

Puis il faut faire une vérification du bon fonctionnement de la carte, si elle est correctement connectée au PC, Pour cela les étapes suivantes sont à suivre :

Vérification du bon fonctionnement de tous les périphériques de la carte DSK :

Double clique sur l'icône « 6416DSK Diagnostics Utility » puis sur le bouton « Start »

IV.4.2 Création d'un projet en « C/C++ » à partir d'un model Simulink

Concernant le passage de Matlab au code composer studio CCS, MATLAB a développé un outil "Link for code composer studio" qui lui permet de communiquer avec le code composer studio et on utilisant cet outil ; le model SIMULINK est traduit en projet écrit en C qui pourra être exécuté sur CCS, pour des applications en temps réel.

Pour démarrer la création du projet en cliquant sur « Incremental build » ou bien sur Ctrl+B, Matlab crée un lien avec le code composer studio et génère le projet écrit en « C ».

```
### Loading TLC function libraries

.....
### Initial pass through model to cache user defined code
.
### Caching model source code
.....
### Writing header file Algorithme_Detection_Tumeur_Cerebrale_types.h
.
### Writing header file Algorithme_Detection_Tumeur_Cerebrale.h
### Writing header file Algorithme_Detection_Tumeur_Cerebrale_private.h
.
### Writing source file Algorithme_Detection_Tumeur_Cerebrale.c
### Writing source file Algorithme_Detection_Tumeur_Cerebrale_data.c
### Writing source file Algorithme_Detection_Tumeur_Cerebrale_main.c
.
### TLC code generation complete.
### Generating the DSP/BIOS configuration file ...
### Creating project in Code Composer Studio(tm)...
### Building Code Composer Studio(tm) project...
### Build complete
### Downloading COFF file
### Downloaded: Algorithme_Detection_Tumeur_Cerebrale.out
```

Figure IV 9 Le passage du Simulink au « C/C++ »

IV.4.4 Différentes opérations pour l'implémentation sur TMS320C6713 DSK

Dans la liste du projet à gauche de la fenêtre du CCStudio, cliquez sur le programme (C/C++) ajouté. Ensuite dans le menu principal cliquez sur « **Project** » et choisissez « **Compile File** ».

Après la compilation du programme, construisez le projet en cliquant dans le menu de « **Project** » puis sur « **Rebuild all** ».

Après la construction du projet, le programme doit être chargé sur la carte. Dans le menu principal choisissez « **File** » et ensuite « **Load Program** ». Une nouvelle fenêtre s'ouvrira dans laquelle vous choisissez le fichier « **Noise_Reducer.out** » et cliquez sur « **Open** ».

Pour exécuter ce programme, cliquez dans le menu « **Debug** » sur « **Go Main** » et ensuite dans le même menu « **Debug** » cliquez sur « **Run** ».

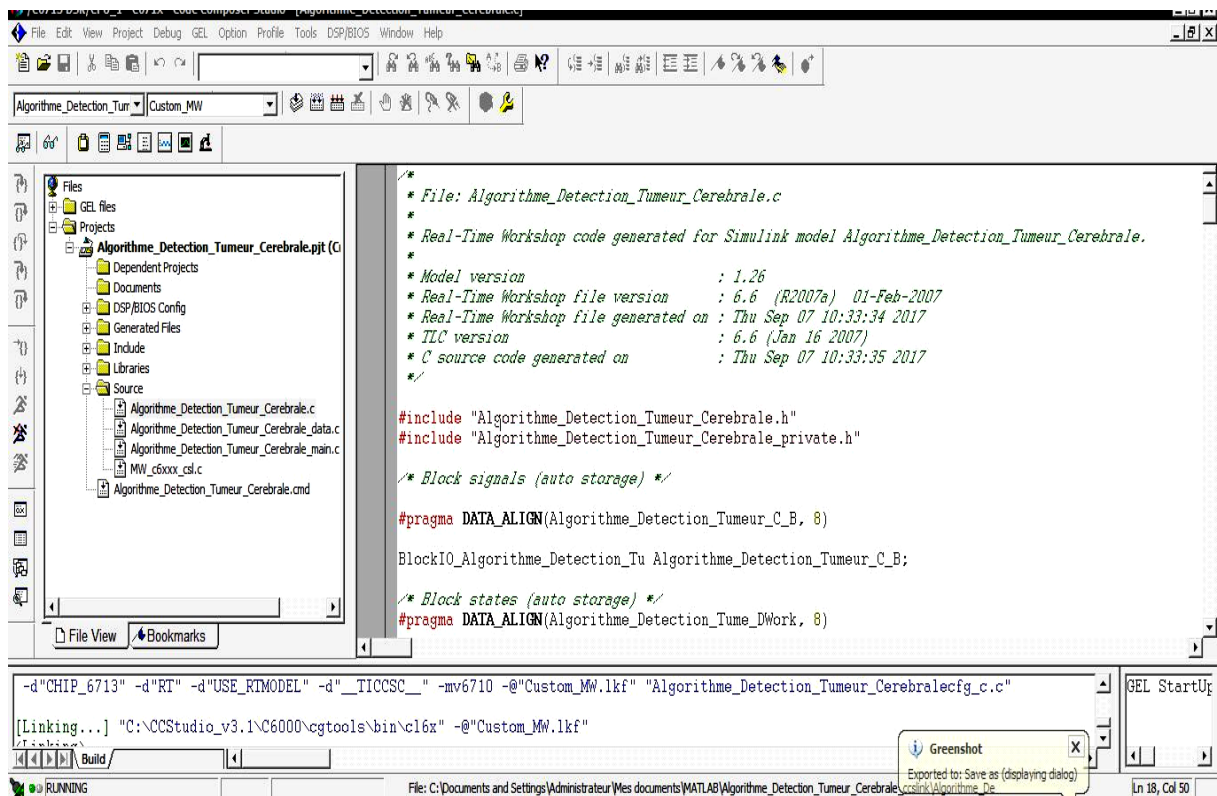


Figure IV 10 Création du projet algorithme_detection_tumeur_cérébrale sur CCS

IV.4.5 Résultats obtenus à l'aide du Code Composer Studio CCS

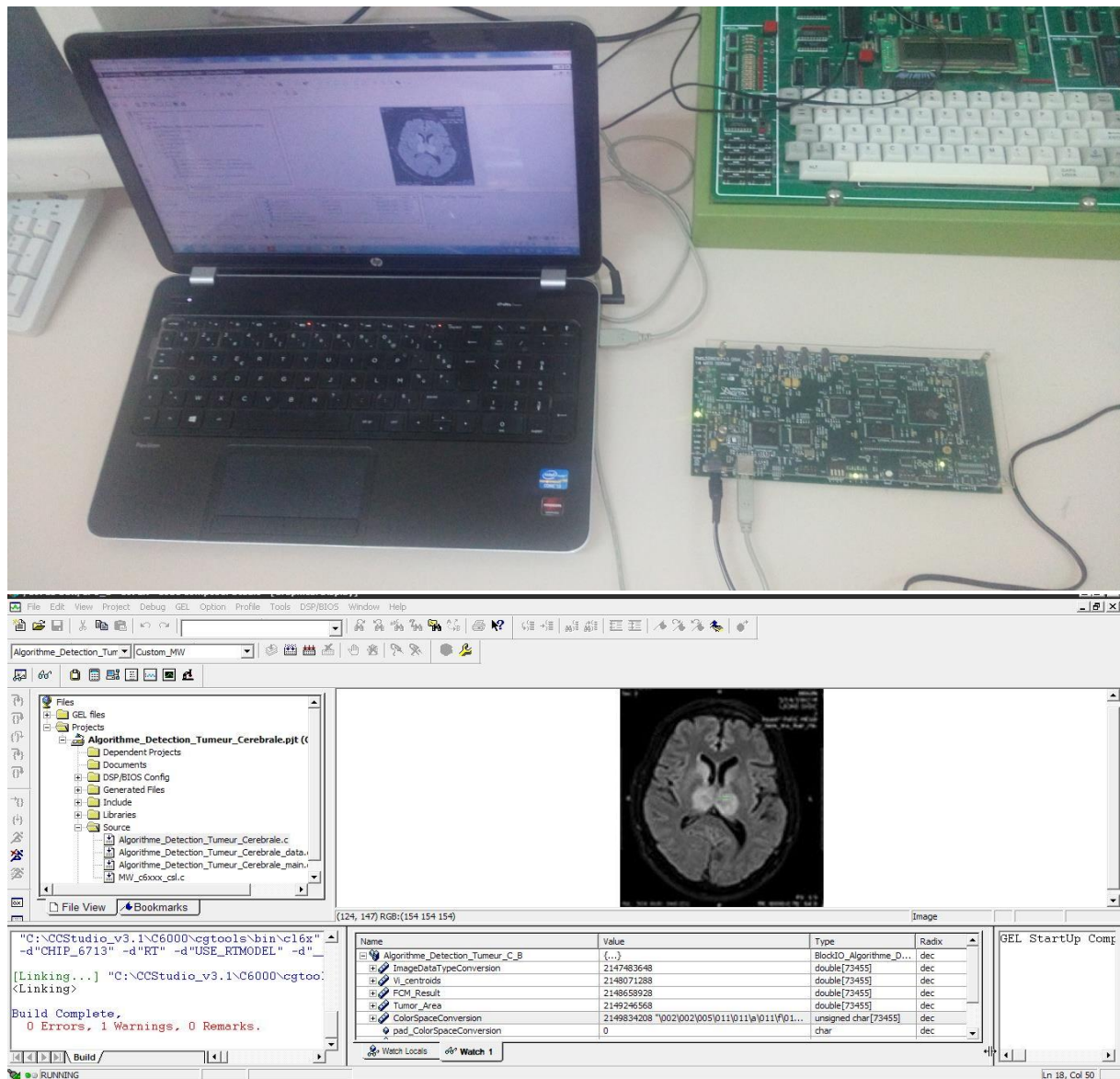


Figure IV 11 Résultats obtenus réellement au labo (INPUT IMAGE)

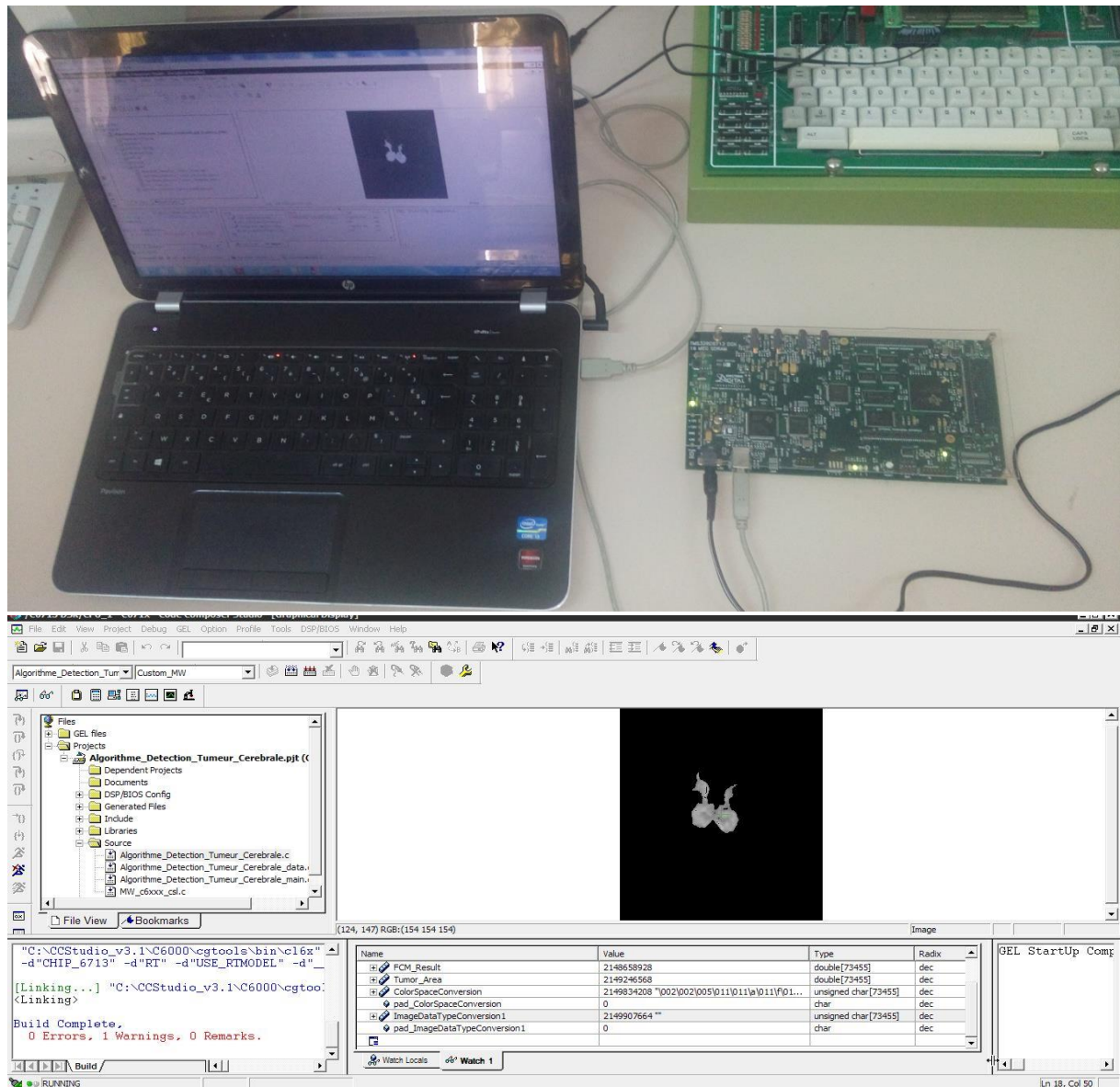


Figure IV 12 Résultats obtenu réellement au labo (OUTPUT IMAGE)

Le tableau ci-dessous c'est un tableau récapitulatif de différentes images IRM traiter, les images INPUT ce sont des image source issues de la base de données **Brainweb**[14], une base de données mise en ligne par le centre d'imagerie cérébrale McConnell de l'institut neurologique de Montréal à l'université McGill. et les images de la case OUTPUT ce son des image segmenter a l'aide de du Code Composer Studio CCS.


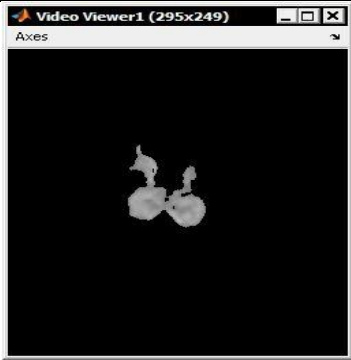






IMAGES	entre	sortie	Taille de tumeur(mm ²)
IMAGE001			11.71
IMAGE002			8.31
IMAGE003			14.91
IMAGE004			17.87



Tableau. IV.2 : les différentes images IRM avant et après segmentation

Interprétation des résultats

D'après les résultats de cette méthode développée, on peut dire que cette méthode donne une bonne segmentation des images IRM cérébrales. L'avantage de cette méthode traduit donc la détection de la partie tumorale et le cerveau. Cette méthode résout le problème de l'initialisation, cette initialisation dépend de plusieurs paramètres comme la largeur, la longueur du contour et le nombre important d'itération. Après l'application de la logique floue (FCM) les objets sont regroupés en classes selon le nombre d'itérations et le degré d'appartenance. L'application de la méthode FCM facilite la détection des régions d'intérêts en utilisant les résultats de k means.

V.3. Inconvénient de la méthode

Dans certains cas, l'algorithme de FCM ne fournit pas une bonne segmentation des tissus cérébraux. L'inconvénient de cette méthode ne tient pas compte que des informations globales de l'image. Dans ce cas, on peut utiliser la croissance de région pour améliorer des résultats et détecter la tumeur cérébrale.

Calcule la taille de tumeur

On voit aussi clairement que la taille de tumeur cérébrale est bien définie, c'est à l'aide du calcul de la zone tumorale et de la détection de scène (**Tumeur Area calculations & Stage détection**), dans cette étape, le calcul de la zone tumorale en utilisant la (4.1) : de binarisation, l'image ayant seulement deux valeurs soit noir (0) soit blanc (1). L'image binaire peut être représentée comme une sommation de nombre total de pixels blancs et noirs [1].

$$\text{Image, } I = \sum_{w=0}^{255} \sum_{H=0}^{255} [f(0) + f(1)] \quad (4.1)$$

Pixels = Largeur (W) x Hauteur (H) = 256 X 256

$f(1)$ = pixel blanc (chiffre 1) et $f(0)$ = pixel noir (chiffre 0)

pixel blanc $P = \sum [f(1)]$ Où P = nombre de pixels blancs (largeur * hauteur)

1 Pixel = 0,264 mm

La formule de calcul de l'aire de la zone tumorale :

$$S = [(\sqrt{P}) * 0.264] \text{ mm}^2 \text{ [15]} \quad (4.2)$$

IV.6 Conclusion

Dans ce chapitre nous avons fait l'implémenté sur une carte de base de DSP TMS320C6416 DSK et Nous avons conclu que le TMS320c6416 est un meilleur choix pour le traitement Numérique d'image. Ceci est traduit par le nombre de cycles d'horloge nécessaire pour effectuer ce traitement.

Conclusion générale

CONCLUSION GENERALE

La segmentation des images médicales reste encore un domaine de recherche très vaste, l'objectif de notre travail est consacré sur la segmentation des tissus cérébraux à partir d'images de résonance magnétique, en vue de segmenter la partie tumorale et aussi le cerveau.

Une étude bibliographique a été faite sur les méthodes de segmentation d'images, nous a permis d'appréhender la diversité des méthodes de segmentation des tissus cérébraux. Plusieurs approches de segmentation sont proposées dans la littérature, la segmentation par contour et la segmentation par région. Dans notre travail, nous avons localisés sur des méthodes utilisées pour la segmentation des structures cérébrales (croissance de région) et les méthodes de classification non supervisés (k-means et Fuzzy CM).

Les résultats trouvés sont satisfaisants, ce qui nous a permis de dire, que l'utilisation d'une méthode combinée entre plusieurs algorithmes de segmentation permet de donner des meilleurs résultats de segmentation.

Enfin, nombreuses images sont segmentées, cette méthode peut être appliquée pour segmenter d'autres structures cérébrales. Cette démarche peut être utilisée pour segmenter d'autres types d'images (images abdomen, images rétiniennes, etc.).

BIBLIOGRAPHIE

- 1 = G.Wyszecki and W.S. Stiles.** Color Science: Concepts and Methods, Quantitative Data and Formulae by Wyszecki, Günther, Stiles, W. S. (2000) Paperback
- 2 = Christine Fernandez-Maloigne :**Image numérique couleur : De l'acquisition au traitement Broché – 2004
- 3 = Christine Fernandez-Maloigne :** Couleur numérique : Acquisition, perception, codage
, 2012
- 4 = Bernd Jähne :**Digital Image Processing,(2 em edition) 2002
- 5 = Rafael C Gonzalez :** Digital Image Processing 3rd Edition (Paperback) Paperback – 2014
- 6 = M. Bosc.** Contribution `a la d´etection de changements dans les s´equences IRM 3D multimodales. PhD thesis, Universit´e Louis-Pasteur-Strasbourg I., 2003
- 7 = J. Sijbers, A. J. den Dekker, P. Scheunders & D. Van Dyck. Maximum Likelihood**
estimation of Rician distribution parameters. IEEE Transactions on Medical Imaging, vol. 17, no. 3,
- 8 = M. A. Gonzalez Ballester, A. P. Zisserman & M. Brady.** Estimation of the partial volume effect. Medical Image Analysis, vol. 6, , 2002.
- 9 = D. W. Shattuck, S. R. Sandor-Leahy, K. A. Schaper, D. A. Rottenberg & R. M. Leahy.** Magnetic Resonance Image Tissue Classification Using a Partial Volume Model.
NeuroImage, vol.
- 10 = B. Belaroussi, J. Milles, S. Carne, Y. M. Zhu & H. Benoit-Cattin.** Intensity non-uniformity correction in MRI : Existing method and their validation. Medical Image Analysis, vol. 10, no. 2, , April 2006.
- 11= C. Jaggi.** Segmentation par m´ethode markovienne de l'enc´ephal humain en imagerie par r´esonance magn´etique : th´eorie, mise en œuvre et ´evaluation. PhD thesis, Universit´e de Caen/Basse-Normandie, 1998.

BIBLIOGRAPHIE

12 = texas instruments inc TMS320C6000 CPU and Instruction Set Reference Guide

Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2010, Texas Instruments Incorporated

13 = **Rulph Chassaing** Digital Signal Processing and Applications with the C6713 and C6416 DSK Worcester Polytechnic Institute

14 = texas instruments inc .TMS320C6414, TMS320C6415, TMS320C6416 Fixed-Point Digital Signal Processors datasheet (Rev. N)

16 =**J.C. Bezdek**, Fuzzy mathematics in pattern classification, PH.D dissertation, Cornell Univ., Ithaca, NY, 1973.

17 =**Dr.S.K.Bandhyopadhyay and Tuhin Utsab Paul**, “ Segmentation of Brain MRI Image – A Review”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012 ISSN: 2277 12

Résumé

Le but de notre travail est de mettre en œuvre une technique de segmentation permettant la détection des tumeurs cérébrales puis l'implémentation de cet algorithme sur une carte base de DSP(TMS3206416DSK). L'objectif de ce mémoire est de concevoir une méthode de classification pour la segmentation d'une image cérébrale .nous avons combiné la méthode c-means pour segmenter les images de cerveau puis on a utilisée la méthode de croissance de région pour détecter et identifier le tumeur cérébrale. Les résultats trouvés sont satisfaisants, ce qui nous a permet de dire, que l'utilisation d'une méthode combinée entre plusieurs algorithmes de segmentation permet de donner des meilleurs résultats de segmentation.

Mots clés : *IRM, segmentation, K-means, FCM, Code Composer Studio, TMS3206416DSK.*

ملخص

الهدف من عملنا هو تطبيق تقنية تجزئة تسمح للكشف عن أورام الدماغ ومن ثم تنفيذ هذه الخوارزمية على بطاقة القائم على معالج الإشارات الرقمية (TMS3206416DSK). الغرض من هذه المذكرة هو تصميم طريقة تصنيف لتجزئة صورة الدماغ، وقد جمعنا طريقة *c-means* لتقسيم صور الدماغ ومن ثم تم استخدام طريقة نمو المنطقة للكشف والتعرف الورم الدماغية. النتائج مرضية، مما يسمح لنا أن نقول، أن استخدام طريقة مجتمعة بين عدة خوارزميات تجزئة يسمح لإعطاء نتائج أفضل للتجزئة.

الكلمات المفتاحية: التصوير بالرنين المغناطيسي، تجزئة، K-means، FCM، Code Composer Studio، TMS3206416DSK.

Abstract

The aim of our work is to implement a segmentation technique allowing the detection of brain tumors and then the implementation of this algorithm on a DSP-based card (TMS3206416DSK).The aim of this paper is to design a classification method for the segmentation of a brain image. We combined the c-means method to segment the brain images and then the region growth method was used to detect and identify the cerebral tumor. The results are satisfactory, which allows us to say, that the use of a combined method between several segmentation algorithms allows to give better results of segmentation.

Keywords : *MRI, segmentation, K-means, FCM, Code Composer Studio, TMS3206416DSK.*