



Université Sidi Mohammed Ben Abdellah
Faculté des Sciences Dhar El Mahraz- Fès
Centre d'Etudes Doctorales
"Sciences et Technologies"

Formation Doctorale : STIC
Discipline : Informatique
Spécialité : Informatique
Laboratoire : LIAN

THÈSE DE DOCTORAT

Présentée par

BENOMAR ELMDEGHRI Saïd

**LA VISUALISATION TRIDIMENSIONNELLE
D'IMAGES MEDICALES A L'AIDE DE LA
TESSELLATION MATERIELLE DES PROCESSEURS
GRAPHIQUES**

Accélération du calcul de rendu et optimisation du stockage

Soutenue le 25 / 11 / 2017 devant le jury composé de :

Pr. Mostafa HARTI	FSDM – Fès	Président
Pr. Mohammed MEKNASSI	FSDM – Fès	Rapporteur
Pr. Brahim OUAHBI	ENSAM – Meknès	Rapporteur
Pr. Mohammed OUMSIS	EST - Salé	Rapporteur
Pr. Mohammed CHERKAOUI MALKI	FMP, Hopital C. Zaid, Rabat	Examineur
Pr. Ismail BERRADA	FSDM – Fès	Examineur
Pr. Mohammed Ouçamah CHERKAOUI MALKI	FSDM – Fès	Directeur de thèse

*La visualisation tridimensionnelle
d'images médicales à l'aide de la
tessellation matérielle des processeurs
graphiques*



Remerciements

Tous d'abord, j'adresse mes plus vifs remerciements et toute ma reconnaissance à mon directeur de thèse : Monsieur le Professeur CHERKAOUI MALKI Mohammed Ouçamah. Sa confiance, ses conseils permanents et ses méthodes scientifiques d'encadrements étaient les points clés d'aboutissement de ce travail.

Mes remerciements s'adressent également à tous les membres du jury qui m'ont honoré d'avoir accepté d'évaluer ce travail. En particulier, je remercie :

- Pr Mostafa HARTI d'avoir accepté de présider le jury de cette thèse.
- Pr. Mohammed MEKNASSI, Pr. Brahim OUAHBI et Pr. Mohamed OUMSIS pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse.
- Pr. Mohammed CHERKAOUI MALKI et Pr. Ismail BERRADA pour l'honneur qu'ils m'ont fait en acceptant d'être examinateurs de cette thèse.

J'implore Allah d'accorder sa miséricorde à notre cher défunt Abdelali ELLOUB, Je tiens à le remercier pour le soutien permanent et la disponibilité qu'il m'a accordé.

Je remercie également les membres de notre équipe de recherche, pour les échanges constructifs que nous avons pu avoir, et pour les moments agréables que nous avons passés ensemble. Je remercie spécialement mon ami Abdelhak KADDARI, pour notre étroite collaboration tout au long de nos études doctorales. Mes remerciements vont également à tous mes amis et mes collègues de travail.

Enfin, je dédie cette thèse à toute ma famille, à ma chère mère à qui je dois toute ma vie, à ma chère femme Fadoua pour son soutien et sa patience, à mes enfants Mouad et Nada, à mes frère Mohamed, Abdeslam et Adil, à ma chère sœur Asmae et tous les membres de ma famille qui n'ont jamais cessé de m'encourager.

De nos jours, La visualisation médicale 3D interactive occupe une place de plus en plus importante dans de nombreuses nouvelles techniques médicales importantes telles que la simulation chirurgicale, la chirurgie et certaines thérapies guidées par image et l'endoscopie virtuelle.

Cependant, le volume des données médicales qui provient des matériels actuels devient de plus en plus important. La reconstruction d'un maillage 3D à partir de ce volume, génère un nombre très important de polygones. Par conséquent, le rendu 3D de ce maillage avec les ordinateurs domestiques ne peut pas satisfaire les exigences de la visualisation 3D, en termes de qualité et d'interactivité. Seules les stations spécifiques – et coûteuses - peuvent satisfaire ces exigences.

Au cours de ce travail, nous nous sommes intéressés à développer des algorithmes permettant d'optimiser le temps d'exécution et l'occupation mémoire de telles applications, et notamment celles qui nécessitent une interactivité avec l'utilisateur (zoom, rotation 3D etc...). Notre but est d'implémenter des algorithmes sur les ordinateurs domestiques équipés par une carte graphique récente.

Notre approche de reconstruction 3D de l'anatomie humaine, à partir d'un volume des données médicales, est basée sur la création d'un maillage basse résolution. Ce maillage de base est obtenu à l'aide de l'algorithme de Marching Cubes. Les petits reliefs perdus lors de la reconstruction du maillage de base sont calculés à partir des informations que nous avons dans les images médicales 2D. Ces reliefs sont représentés sous forme de vecteurs 2D, ce qui permet d'avoir une optimisation très importante au niveau de stockage de ce maillage 3D. Les détails de chaque polygone du maillage de base sont calculés d'une manière indépendante, cela nous facilite la génération automatique de la carte du déplacement pour ce maillage. L'utilisation de cette carte de déplacement avec les GPU récents permet d'accélérer les calculs du rendu et d'avoir une visualisation en temps réel.

Abstract

Today, Interactive 3D medical visualization is becoming increasingly important in many important new medical techniques such as surgical simulation, surgery guided by image and virtual endoscopy.

However, the volume of medical data acquired from existing materials becomes more important. The reconstruction of a 3D mesh from this volume generates a very large number of polygons. Consequently, the 3D rendering of this mesh with personal computers can not satisfy the requirements of 3D visualization, in terms of quality and interactivity. Only specific - and expensive - stations can satisfy its requirements.

In this work, we are interested in developing algorithms to optimize the execution time and memory occupancy of such applications, and in particular those which require interactivity with the user. Our goal is to implement its algorithms on personal computers equipped with a recent graphics card.

Our 3D reconstruction approach to human anatomy, from a volume of medical data, is based on the creation of a low-resolution mesh. This basic mesh is obtained using the Marching Cubes algorithm. The small reliefs lost during the reconstruction of the basic mesh are calculated from the information we have in the 2D medical images. These reliefs are represented as 2D vectors, which provides a very important optimization in storage this 3D mesh. The details of each polygon of the basic mesh are computed independently, which facilitates the automatic generation of the displacement map for this mesh. The use of this card with the recent GPU allows to accelerate the rendering calculations to have a visualization in real time.

Table des matières

TABLE DES MATIÈRES	V
LISTE DES FIGURES	VIII
LISTE DES SIGLES	XI
1. INTRODUCTION GÉNÉRALE	12
1.1. INTRODUCTION.....	13
1.2. CADRE DE LA THÈSE	13
1.3. MOTIVATIONS ET BUT.....	15
1.4. CONTRIBUTIONS	17
1.5. ORGANISATION DU DOCUMENT	18
2. IMAGERIE MÉDICALE	20
2.1. INTRODUCTION.....	21
2.2. MODALITÉS D'IMAGERIE.....	22
2.2.1. <i>Tomodensitométrie</i>	22
2.2.1.1. Principe physique du TDM	23
2.2.1.2. Restitution d'une image	27
2.2.1.3. Utilisation de la tomodensitométrie	31
2.2.2. <i>Imagerie par résonance magnétique</i>	32
2.2.2.1. Principe Physique de la Résonance Magnétique	33
2.2.2.2. Construction d'une image en Résonance Magnétique	35
2.2.2.3. Utilisation de l'Imagerie par Résonance Magnétique	38
2.2.3. <i>Imagerie en médecine nucléaire</i>	40
2.2.3.1. Tomographie par émission de positons	40
2.2.3.2. Tomographie par émission mono photonique.....	42
2.3. TRAITEMENT ET ANALYSE DES IMAGES MÉDICALES	44
2.3.1. <i>Recalage</i>	45
2.3.2. <i>Segmentation</i>	47
2.3.2.1. Exigences de la segmentation	48
2.3.2.2. Segmentation manuelle	49
2.3.2.3. Segmentation basée sur un seuil	50
2.3.2.4. Région croissance.....	50
2.3.2.5. Segmentation de Watershed	51

2.3.3.	<i>Visualisation</i>	54
3.	RENDU DE SURFACE	57
3.1.	INTRODUCTION.....	58
3.2.	PROBLÈMES TOPOLOGIQUES	60
3.3.	REPRÉSENTATION DE MAILLAGES SURFACIQUES	62
3.3.1.	<i>Représentation des vertex partagés</i>	62
3.3.2.	<i>Structure de données Arête-Ailée</i>	64
3.3.3.	<i>Structure des données Demi-arêtes</i>	65
3.4.	DICOM ET LE STOCKAGE DU MAILLAGE.....	66
3.4.1.	<i>Standard DICOM</i>	66
3.4.2.	<i>Structure des données DICOM</i>	67
3.4.3.	<i>Supplément DICOM de description polygonal d'une surface</i>	69
3.5.	MARCHING CUBES.....	71
3.5.2.	<i>Marching squares</i>	73
3.5.3.	<i>Algorithme de base 3D</i>	74
3.5.3.1.	Détermination de l'indice de cas et des arêtes d'intersection.....	75
3.5.3.2.	Calcul des intersections	77
3.5.3.3.	Triangulation des Intersections.....	77
3.5.4.	<i>Limites et améliorations de Marching cubes</i>	78
3.5.4.1.	Problèmes de topologie	79
3.5.4.2.	Ambiguïtés en 3D	80
4.	UNITÉ DE TRAITEMENT GRAPHIQUE	83
4.1.	INTRODUCTION.....	84
4.2.	SHADERS.....	86
4.2.1.	<i>Programmation des cartes graphiques</i>	86
4.2.2.	<i>Shaders de bas niveau</i>	87
4.2.3.	<i>Shaders de haut niveau</i>	88
4.3.	PIPELINE GRAPHIQUE DE RENDU.....	89
4.3.1.	<i>Processeur graphique</i>	89
4.3.2.	<i>Les étapes de pipeline graphique</i>	92
4.4.	TESSELLATION.....	97
4.4.1.	<i>Principe de tessellation</i>	97
4.4.2.	<i>Étapes de tessellation matérielle</i>	98

4.4.2.1.	Étape Hull shader	99
4.4.2.2.	Tessellator étape.....	100
4.4.2.3.	Domain shader étape.....	101
4.4.3.	<i>Displacement Mapping</i>	102
5.	RECONSTRUCTION D'UN MAILLAGE 3D AVEC VECTEURS DE DEPLACEMENTS POUR LES IMAGES MEDICALES	105
5.1.	INTRODUCTION.....	106
5.2.	MÉTHODOLOGIE.....	106
5.2.1.	<i>Vue d'ensemble du nouvel algorithme</i>	107
5.2.2.	<i>Application de Marching Cubes Adaptative</i>	109
5.2.3.	<i>Modification du maillage basse résolution</i>	110
5.2.4.	<i>Extraction des vecteurs de déplacement de chaque polygone</i>	113
5.2.5.	<i>Génération automatique de la carte de déplacement</i>	116
5.3.	RÉSULTATS & DISCUSSIONS	118
5.4.	CONCLUSION	126
6.	AMELIORATION DE L'EXTRACTION DES VECTEURS DE DEPLACEMENT POUR UN MAILLAGE 3D DES IMAGES MEDICALES.....	127
6.1.	INTRODUCTION.....	128
6.2.	MÉTHODOLOGIE.....	130
6.2.1.	<i>Vue d'ensemble de l'algorithme d'extraction</i>	130
6.2.2.	<i>Extraction du contour</i>	132
6.2.3.	<i>Discretisation des points du contour</i>	135
6.3.	RÉSULTATS & DISCUSSIONS	137
6.4.	CONCLUSION	141
7.	CONCLUSION GÉNÉRALE	143
	BIBLIOGRAPHIES	147

Liste des figures

Figure 1-1 : Comparaison entre la représentation 2D et 3D des images médicales.	15
Figure 2-1 : Principe de la réalisation d'une coupe (image 2D) par tomодensitométrie.	24
Figure 2-2 : Interpolation des valeurs des voxels dans une coupe TDM.	25
Figure 2-3 : Fonctionnement général des différentes générations de scanners.	26
Figure 2-4 : Une photo d'un scanner moderne (Siemens, Somatom Sensation).	27
Figure 2-5 : Une coupe de TDM où les implants métalliques dans les dents entravent fortement le diagnostic.	30
Figure 2-6 : Visualisation d'une image TDM après injection de produit de contraste.	32
Figure 2-7 : Représentation du spin de particules soumises à un champ magnétique.	33
Figure 2-8 : Effet d'une impulsion électromagnétique.	34
Figure 2-9 : Relaxation après impulsion.	35
Figure 2-10 : Gradient magnétique généré dans les trois axes de l'espace.	37
Figure 2-11 : IRM cérébrale en pondérations T1, T2.	38
Figure 2-12 : Des images reconstruites à partir du cerveau (IRMf) Tout en montrant les lettres correspondantes	39
Figure 2-13 : La détection de l'annihilation en TEP.	41
Figure 2-14 : La représentation des données TDM et TEP.	42
Figure 2-15 : Hybrid scanner TEMP / TDM, GE NM / CT 670 de General Electric.	44
Figure 2-16 : Illustration de recalage de deux modalités complémentaires.	46
Figure 2-17 : Les différentes étapes d'un algorithme de région croissance.	51
Figure 2-18 : Principe de la transformation du WaterShed où les valeurs d'intensité définissent les collines et les bassins	53
Figure 2-19 : Illustration du problème de sur-segmentation de la transformation Watershed appliqué à une coupe axiale d'une image TDM.	53
Figure 2-20 : L'utilisation des marqueurs avec Watershed.	54
Figure 2-21 : Les voies neurales sont simulées à partir d'images de tenseur de diffusion	55
Figure 2-22 : la visualisation de volume interactif d'une angiographie TDM au diagnostic d'un accident vasculaire cérébral ischémique et de planifier la chirurgie reconstructive	56
Figure 2-23 : Une vue démêlée du côlon utilisé pour trouver rapidement des polypes dans la coloscopie TDM virtuelle	56

Figure 3-1 : Comparaison entre le rendu surfacique et volumétrique au niveau d'éclairage	59
Figure 3-2 : Trois solutions possibles au problème de la correspondance en cas de changement de topologie.	61
Figure 3-3 : Problème de triangulation de contours de deux coupes.	61
Figure 3-4 : structure de données Arête-Aillée.	64
Figure 3-5 : Demi-arête et relations d'incidence stockées.	65
Figure 3-6 : Représentation du modèle DICOM et des étapes médicales équivalentes.	68
Figure 3-7 : Maillage de Tétraèdre.	70
Figure 3-8 : Extraction de l'isosurface à l'aide de Marching cubes en traitant localement les cellules de chaque coupe.	72
Figure 3-9 : les contours possibles d'un ensemble de pixels.....	73
Figure 3-10 : Indexation des Voxels et des arêtes de l'algorithme Marching Cubes.	75
Figure 3-11 : 15 classe des cas de Marching Cubes.....	76
Figure 3-12 : Iso-surfaces extraites à partir des données TDM avec Marching Cubes	78
Figure 3-13 : Un résultat de Marching Cubes à partir d'une segmentation d'un vaisseau sanguin allongé, partiellement fortement courbées.....	79
Figure 3-14 : Les triangulations possibles de Marching Cubes pour deux cellules de volumes voisins.....	80
Figure 4-1 : l'évolution de la puissance de calcul des CPUs et GPUs.....	85
Figure 4-2 : Les shaders sont souvent utilisés pour produire l'éclairage et l'ombre dans la modélisation 3D.....	87
Figure 4-3 : Les dix types de primitives définies dans OpenGL.	89
Figure 4-4 : Communication entre le CPU, le GPU et les mémoires associées.	91
Figure 4-5 : Diagramme du pipeline graphique de rendu programmable	93
Figure 4-6 : Pipeline graphique selon la nomenclature DirectX 11 comportant les étapes shaders programmables et matériels configurables.....	98
Figure 4-7 : Création des modèles de tessellation d'un quad pour différents facteurs de tessellation.	100
Figure 4-8 : Création des modèles de tessellation d'un triangle pour différents facteurs de tessellation.	101
Figure 4-9 : Comparaison entre le Bump Mapping et le Displacement Mapping	102

Figure 4-10 : Rendu de "Unigine Heaven Benchmark" avec et sans l'utilisation de tessellation matérielle.	103
Figure 4-11 : Exemple de l'utilisation de tessellation matérielle pour le rendu d'une surface avec des petites reliefs.	104
Figure 5-1 : schéma représentant les différentes étapes de notre algorithme.	108
Figure 5-2 : Fissure sur une face commune entre les cellules de subdivision différente	110
Figure 5-3 : Modification du maillage.....	111
Figure 5-4 : La suppression des sommets entre les images.	112
Figure 5-5 : Transformation des polygones triangulaire en quadrilatère.	112
Figure 5-6 : La perte des informations dans le maillage de base.	113
Figure 5-7 : L'extraction des détails perdus.	114
Figure 5-8 : L'interpolation des points de surface à l'aide de deux courbes $V1(u)$ et $V2(u)$	117
Figure 5-9 : Structure de la carte de déplacement.	117
Figure 5-10 : l'aspect de « marches d'escalier » sur les surfaces obtenu par le Marching Cubes.	118
Figure 5-11 : Comparaison de la qualité du maillage obtenu avec notre méthode et le maillage de Marching Cubes	118
Figure 5-12 : Le maillage tesselé obtenu à partir le maillage de base	119
Figure 5-13 : Représentation d'un polygone quadrilatère [abcd] du maillage de base avec ces vecteurs de déplacement.....	122
Figure 5-14 : Le rendu classique et le rendu avec tessellation matérielle.	125
Figure 6-1 : hétérogénéité des détails extrait de chaque côté.	129
Figure 6-2 : Erreurs d'extraction des petits reliefs.	130
Figure 6-3 : Les entrées de l'algorithme d'extraction des vecteurs de déplacement : Maillage de base , images médicales segmentées	132
Figure 6-4 : homogénéité au niveau de détailles extrait de chaque polygone.	137
Figure 6-5 : Extraction du contour avec les vecteurs de déplacement.	138
Figure 6-6 : comparaison entre les deux méthodes d'extraction.	139

Liste des sigles

CMS	: Cubical Marching squares
CPU	: Central processing unit
DC	: Dual contouring
DICOM	: Digital Imaging and Communications in Medicine
DTI	: Diffusion tensor imaging
EMC	: Extended Marching Cubes
Flops	: floating-point operation per second
GPU	: Graphics Processing Unit
HAL	: Hardware Abstraction Layer
HU	: Hounsfield
IRM	: Imagerie par résonance magnétique
MC	: Marching cubes
NEMA	: National Electrical Manufacturers Association
TCS	: Tessellation Control Shader
TDM	: Tomodensitométrie
TEMP	: Tomographie par émission mono photonique
TEP	: Tomographie par émission de positons
TES	: Tessellation Evaluation Shader
VRAM	: Video Random Access Memory

1. Introduction générale

1.1. Introduction

Dans cette thèse, nous présentons une nouvelle méthode qui exploite les progrès récents dans la puissance de calcul et la programmabilité de certaines unités des cartes graphiques afin d'accélérer la visualisation 3D des données médicales. Notre méthode permet aussi de diminuer la quantité des informations à stocker pour le modèle 3D de l'organe anatomique concerné.

Dans ce chapitre, nous donnons un aperçu succinct de l'imagerie médicale selon une optique qui illustre comment les changements dans le matériel informatique ont influencé la visualisation médicale. Nous présentons ensuite les motivations de notre recherche et ses objectifs. Nous y faisons l'annonce de nos contributions et fournissons au lecteur l'organisation générale de cette thèse.

1.2. Cadre de la thèse

En médecine moderne, l'imagerie prend de plus en plus de place. Elle est de nos jours quasiment incontournable dans toutes les pratiques quotidiennes. L'imagerie médicale se réfère à un ensemble de techniques qui produisent des images pour les parties internes du corps. La technologie d'imagerie a grandement évolué, ce qui permet d'avoir une amélioration très importante au niveau de la visualisation interne du corps selon une séquence de coupes 2D. Cette technologie est maintenant utilisée non seulement dans les procédures de diagnostic, mais aussi dans la planification, l'évaluation et l'assistance de la chirurgie et dans la radiothérapie.

L'imagerie médicale ne se limite pas seulement à la lecture d'appareils de mesure qui déterminent l'interaction physique entre l'énergie et la matière (cf. chapitre suivant), mais s'appuie aussi sur les technologies qui permettent d'acquérir, traiter et afficher d'énormes quantités de données biologiques.

Dans un passé proche l'imagerie médicale faisait principalement référence aux rayons X. Ces rayons qui traversent le corps et fournissent, en résultat, un film semblable

à une photographie. Au cours des quatre dernières décennies, l'imagerie médicale a progressé de manière vélocité, soutenue par l'énorme progrès de la technologie, entre autre, numérique. Elle s'est toutefois projetée de devenir moins coûteuse, plus compacte et plus rapide au niveau des calculs.

La visualisation médicale est le processus de la génération d'images à partir de données physiques brutes afin de mieux comprendre les caractéristiques médicales qualitatives et quantitatives.

En général, les images générées sont 2D et les cliniciens peuvent avoir un aperçu de la visualisation de données 3D comme une séquence de coupes 2D. La reconstruction 3D des données fournit une vue beaucoup plus riche et très complète des structures contenues dans l'image, en raison de notre capacité implicite de percevoir des formes en 3D (voir Figure 1-1).

Au cours de ces quatre décennies, les modalités d'imagerie telles que la tomodensitométrie (TDM), la résonance magnétique (IRM), la tomographie par émission de positons (TEP) et les ultrasons ont été développés pour obtenir non seulement des images projectives mais aussi des images en coupe transversale du corps. Contrairement aux images projectives qui projettent un volume 3D entier dans une image 2D, une image transversale permet d'obtenir une vue sans occlusion d'une coupe 2D du corps. En outre, par l'acquisition de nombreuses coupes 2D, les images 3D complètes peuvent désormais être systématiquement acquises dans toutes les modalités d'imagerie mentionnées ci-dessus.

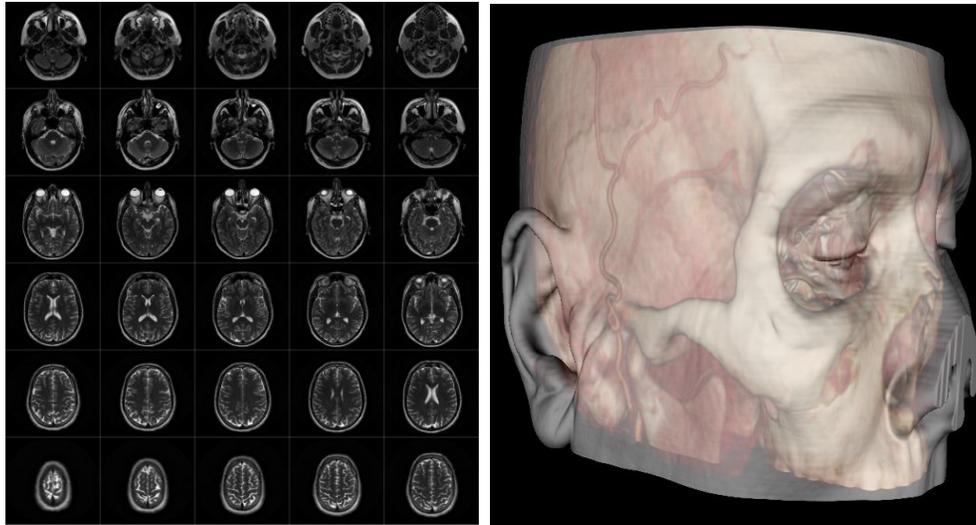


Figure 1-1 : Comparaison entre la représentation 2D et 3D des images médicales.

*Note : À gauche : Une partie d'une série des coupes 2D à partir d'un scanner IRM.
À droite : Le même scan rendu en utilisant la visualisation 3D.*

1.3. Motivations et But

Le plus grand obstacle de la visualisation médicale 3D a longtemps été celui du matériel utilisé. En raison des grandes exigences de calcul d'un ensemble de données 3D, une seule image pourrait être générée toutes les quelques secondes ou quelques minutes en utilisant des algorithmes de rendu hors ligne.

La lenteur du processus l'emportait souvent sur son utilité, en particulier lorsqu'on a de nombreux patients en attente de diagnostic ou de traitement en temps utile. Les nombreuses techniques logiciels de reconstruction qui visaient (aussi) une interactivité acceptable, peuvent malheureusement aussi réduire la précision du rendu, et cette diminution de la précision pourrait conduire à des erreurs médicales (GREMSE, et al., 2016).

Afin d'améliorer l'interactivité de la visualisation médicale 3D, des solutions «matériel» ont été développées. Elles consistent en des stations de travail personnalisées - et coûteuses- associées avec un matériel dédié qui permet l'interactivité en temps réel de la visualisation 3D. Ces stations de travail ont d'ailleurs donné le jour à de nouvelles techniques telles que la simulation chirurgicale, la chirurgie guidée par l'image et l'endoscopie virtuelle (J. Sobocinski, et al., 2013). Ces techniques ont malheureusement un très grand coût et sont extrêmement rigides par rapport aux logiciels de l'ordinateur standard.

Dans les quelques dernières années, un changement majeur est survenu dans la conception de processeur de l'ordinateur. Les processeurs ont atteint les limites pratiques de la vitesse d'horloge, et ont évolué vers des processeurs parallèles (multi-core) pour réaliser de nouveaux gains de vitesse. Ce changement a été alimenté principalement par les processeurs graphiques. Ces processeurs ont été initialement conçus pour soutenir le processeur principal en traitant pour lui et en parallèle certaines tâches d'affichage. Cela pour résoudre la tâche de rendre des millions de polygones dans les jeux vidéo. Comme les processeurs graphiques sont devenus plus flexibles, il est devenu évident qu'ils peuvent être utilisés non seulement pour rendre des polygones, mais aussi pour effectuer une variété de tâches de traitement « des données parallèles », et que les algorithmes conçus pour une telle architecture peuvent servir à une variété de tâches de calcul haute performance.

Parmi les principaux objectifs de cette thèse, nous démontrons l'intérêt de l'utilisation de la carte graphique moderne afin de réduire de manière efficace, le temps d'exécution et d'augmenter la qualité de la visualisation médicale 3D. Nous présentons une nouvelle méthode qui exploite les avancées récentes dans la puissance et la programmabilité des unités de traitement de la carte graphique, afin d'accélérer grandement la visualisation médicale 3D tout en augmentant simultanément la qualité d'image de diagnostic et d'optimiser le stockage de ces modèles 3D.

Il faut préciser que dans ce contexte, il y a trois contraintes interdépendantes qui doivent être prises en compte lors de la conception d'une application de visualisation médicale :

- Diagnostic de qualité : le résultat final obtenu doit contenir tous les détails utiles de l'objet 3D à visualiser.
- Interactivité : le nombre des images calculé par seconde lors de la visualisation doit être suffisant pour donner une interactivité en temps réel.
- Réalisme : Le calcul de l'éclairage peut fournir des détails supplémentaires et des indices de profondeur.

1.4. Contributions

Nos contributions sont directement liées à l'objectif que nous nous sommes fixé ci-dessus, à savoir : conserver toutes les informations utiles dans la visualisation 3D de l'organe anatomique, sans ralentir la vitesse du rendu.

Notre première contribution consiste à accélérer les calculs d'affichage pour avoir une visualisation 3D en temps réel. En effet, les techniques de visualisation surfacique des données médicales exigent de déterminer le niveau de détails qui sera affiché lors de la visualisation 3D, la conservation de toutes les informations utiles contenues dans les images médicales 2D conduit automatiquement à un grand ralentissement dans le rendu. Ce ralentissement étant dû à la grande quantité des données qui doivent être traitées. Dans notre travail, nous nous sommes basées sur l'algorithme de Marching cubes adaptatifs avec une résolution très basse afin de générer seulement la forme de base de l'organe qui sera visualisé. Et à l'aide de notre nouvel algorithme, nous générons une carte de déplacement qui contient tous les détails perdus dans la forme de base. Cette carte de déplacement sera exploitée directement par le processeur de la carte graphique afin

d'ajouter ces détails. Sachant que la puissance de calcul du GPU¹ est beaucoup plus grande que celle du CPU², cela permet d'accélérer les calculs d'affichage pour avoir une visualisation 3D fluide de l'organe anatomique.

Notre deuxième contribution concerne le stockage du maillage utilisé dans la visualisation 3D de l'organe anatomique. Le stockage du résultat de la reconstruction 3D permet d'éviter de réexécuter les différentes étapes qui permettent la visualisation 3D (recalage, segmentation, rendu ...)., Notre politique de stockage permet, d'autre part, de visualiser le résultat obtenu dans les autres postes qui ne sont pas nécessairement équipés de logiciels adéquats pour la reconstruction 3D. Le résultat stocké peut être visualisé tout simplement avec d'autres applications qui permettent l'affichage des maillages 3D.

Par ailleurs, notre méthode de stockage représente les petits reliefs du maillage de base sous forme des vecteurs 2D, cela conduit à une optimisation très importante au niveau du stockage final par rapport aux méthodes classiques de stockage.

1.5. Organisation du document

Dans le chapitre 2, nous donnons une vue globale sur l'imagerie médicale. Nous fournissons au début un aperçu sur l'acquisition d'un volume de données médicales selon différentes modalités. Puis nous présentons les trois axes de base dans les traitements d'images médicales (le recalage, la segmentation et la visualisation) qui sont essentielles pour appréhender ce travail de thèse.

¹ *Graphics Processing Unit*

² *Central processing unit*

Étant donné que cette thèse se concentre sur la visualisation 3D d'un volume de données médicales dans le cadre du rendu de surface, le chapitre 3 est consacré à la représentation de ce type de rendu, nous commençons par la présentation des problèmes topologiques que nous pouvons avoir durant la reconstruction de la surface externe d'un objet 3D, puis nous illustrons le principe du Marching Cubes en fournissant son algorithme de base pour l'extraction d'une isosurface à partir d'un volume de donnée. Nous discutons ensuite les limites et les améliorations qui ont été apportées à cet algorithme.

Dans le chapitre 4, nous présentons les processeurs graphiques modernes, nous expliquons l'architecture de pipeline graphique et en particulier la nouvelle unité de tessellation qui est en relation direct avec nos contributions. Nous y présentons aussi la technique de « displacement mapping » qui permet de profiter de cette nouvelle unité dans notre application.

Dans le chapitre 5, nous présentons notre nouvelle méthode pour la visualisation 3D d'un volume de données médicales à l'aide de l'unité de tessellation du GPU. Nous commençons par une description générale de cette méthode, puis nous expliquons en détails les différentes étapes de notre approche de visualisation : construction du maillage de base, extraction des vecteurs de déplacement et génération automatique de la carte de déplacement. La dernière partie de ce chapitre est consacrée à la présentation des résultats obtenus par notre méthode, notamment en les comparant avec la technique de rendu classique.

Dans le chapitre 6, nous présentons une amélioration de l'algorithme utilisé dans le chapitre précédant pour l'extraction des vecteurs de déplacement. Puis nous montrons l'intérêt de cette amélioration pour surpasser certains problèmes détectés avec notre premier algorithme d'extraction.

Le chapitre 7 conclut la thèse en donnant un bilan du travail effectué et les perspectives envisagées au terme de cette thèse.

2. Imagerie médicale

Ce chapitre présente une vue globale sur l'acquisition, le traitement et l'analyse des images médicales. Ces méthodes sont en relation directe avec le sujet de notre thèse. Ce chapitre ne vise pas à fournir un état de l'art détaillé, il cherche plutôt à mettre le lecteur dans notre contexte de recherche et à fournir un aperçu général sur l'imagerie médicale pour l'informaticien intéressé n'ayant pas nécessairement des connaissances profondes dans ce domaine.

Nous l'entamons avec l'imagerie médicale, suivie d'une description des modalités d'imagerie médicale actuelle. Puis nous fournissons un aperçu sur les trois piliers de l'imagerie médicale qui sont le recalage, la segmentation et la visualisation.

2.1. Introduction

L'imagerie médicale est souvent utilisée pour désigner l'ensemble des techniques qui permettent aux cliniciens de découvrir l'intérieur du corps humain sans intervenir chirurgicalement. La première technique utilisée a été celle des Rayon-X. Cette technique a largement servi cet objectif depuis plus d'un siècle. Cependant, au cours des dernières décennies, il y a eu une explosion de nouvelles techniques d'imagerie qui peuvent fournir une gamme d'informations essentielles sur les propriétés structurales, chimiques et électriques du corps humain.

Les techniques d'imagerie médicale peuvent être considérées comme solutions des problèmes inverses en mathématiques. Cela signifie simplement que la cause (propriétés du tissu vivant) doit être déduite de l'effet (les mesures observées à partir de l'appareil d'imagerie). Les problèmes de cette nature ont souvent de nombreuses solutions valables. Ceci signifie que si les solutions peuvent fournir des informations cruciales, il y aura toujours un certain niveau d'incertitude. Il est donc souvent nécessaire d'une interprétation humaine des résultats.

Mais alors que le corps est intrinsèquement 3D, les images capturées par des radiographies étaient toujours 2D, ce qui entraîne l'ambiguïté. La solution à ce problème est venue au début des années 80 où les avancées en ingénierie et en informatique ont apporté des nouvelles technologies d'imagerie 3D telles que la tomодensitométrie (TDM), l'imagerie par résonance magnétique (IRM), la tomographie par émission de positons (TEP) et l'échographie 3D.

La technologie derrière chacune de ces approches est différente, ce qu'ils ont en commun c'est la capacité de capturer des images en coupe, plutôt que plan-projection. Alors que dans une coupe, qui est également 2D, chaque pixel de l'image représente une position « point » unique dans le corps, dans une image projective le pixel représente une ligne à travers le corps. Ainsi, en capturant de nombreuses coupes 2D, un volume 3D complet peut être obtenu. Les pixels de chaque coupe deviennent voxels du volume résultant en relation directe avec des emplacements dans le corps.

Cependant, bien que ces nouvelles technologies d'imagerie puissent fournir une grande quantité d'informations sur un patient, l'interprétation de ces informations n'est pas simple. Chaque modalité d'imagerie profite de différentes propriétés physiques pour créer une image, et dans la forme brute, ces données peuvent être difficiles à déchiffrer. Dans le cas de l'échographie, l'image se compose d'écho des ondes de pression sonores, tandis que dans la projection de radiographie (rayons X), la mesure est la force des rayons X atténués après avoir parcouru des matériaux de différentes densités atomiques. Dans chaque cas, ce qui est enregistré par le dispositif d'imagerie et ce qui est compris par le clinicien ne sont pas complètement alignés (Badano, et al., 2015).

Les logiciels informatiques peuvent grandement aider à transformer les données de leur forme brute en une forme qui est plus lisible pour un observateur humain. Ce qui permet d'aligner ainsi les exigences disparates des médecins avec le produit obtenu par les méthodes d'acquisition d'imagerie. Pour cette raison, l'acquisition de l'image physique et le traitement de l'image sur ordinateur sont souvent étroitement liés dans ce cadre de l'imagerie médicale.

2.2. Modalités d'imagerie

En imagerie médicale, il existe plusieurs types d'équipements utilisés pour acquérir des images du corps, telles que la radiographie, l'échographie et l'imagerie par résonance magnétique. Chaque type d'acquisition est appelé modalité. Chaque modalité est donc basée sur des phénomènes physiques différents et permet de capturer des types d'informations différentes. Les informations capturées sur le patient selon différentes modalités peuvent être complémentaires. Une modalité suffit généralement pour diagnostiquer un problème donné, mais dans certaines situations, plusieurs modalités peuvent être sollicitées pour un seul cas.

2.2.1. Tomodensitométrie

La tomodensitométrie (TDM), est une technique qui permet l'acquisition des images médicales sous forme des coupes 2D, chaque image représente l'atténuation d'un

faisceau de rayons X dans un volume étudié. La TDM est une méthode qui détermine les valeurs des voxels à l'intérieur d'un volume des données en se basant sur une séquence d'images 2D acquises sous différents angles.

Dans cette section, nous représentons le fonctionnement général d'un appareil TDM, nous commençons par le principe physique de cette technique, puis la restitution de l'image et finalement son utilisation médicale.

2.2.1.1. Principe physique du TDM

Selon un plan prédéterminé, une source X envoie un ensemble de faisceaux de rayon X. Sur chacune des trajectoires de ces faisceaux, on dispose d'un ensemble des détecteurs pour mesurer l'atténuation des rayons X (voir Figure 2-1). A l'aide des résultats fournis par les différents faisceaux, on calcule le coefficient d'atténuation de rayon X de la zone à étudier.

L'intensité I mesurée par un détecteur d'un faisceau de rayon X, est décrite par la formule suivante :

$$I_x = I_0 \cdot e^{-\mu x}$$

I_0 : l'intensité initiale du faisceau de rayons X.

x : l'épaisseur du matériau traversé.

μ : le coefficient d'absorption linéaire du matériau traversé.

I_0 et x sont des données déjà connues et en mesurant I , on peut déduire le coefficient d'absorption, $\mu = \log \left(\frac{I_x}{I_0} \right)$.

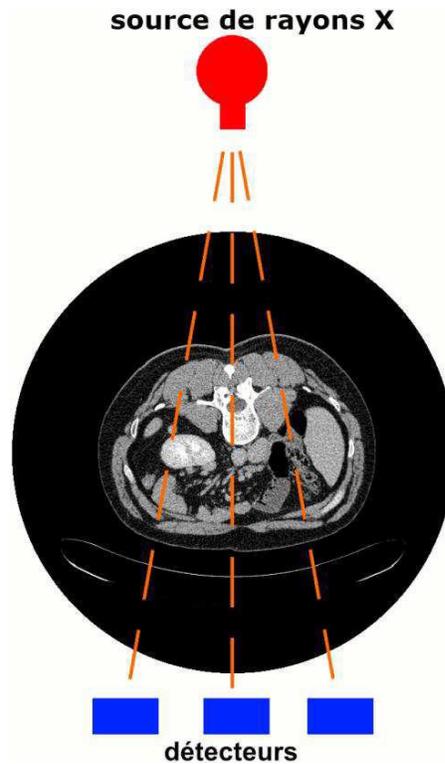


Figure 2-1 : Principe de la réalisation d'une coupe (image 2D) par tomодensitométrie.

Sur le même plan, on réalise un grand nombre de mesure avec des angles d'incidence différents. Pour chaque mesure, on détermine le coefficient d'atténuation de rayon X. Par interpolation, il est possible d'associer à chaque unité de volume étudié (appelé voxel) un coefficient d'atténuation.

A l'aide de la transformation inverse de Radon, on peut reconstruire une image concernant le plan étudié. La Figure 2-2 montre le principe général de la méthode de reconstruction d'une coupe. Dans cet exemple, on a des projections du rayon X selon quatre incidences. La partie gauche présente la grille de pixels correspondant au volume étudié et donne aussi la valeur d'atténuation de chaque projection. Pour la partie droite, on a le résultat obtenu de la reconstruction d'image après l'application de la transformation inverse de Radon.

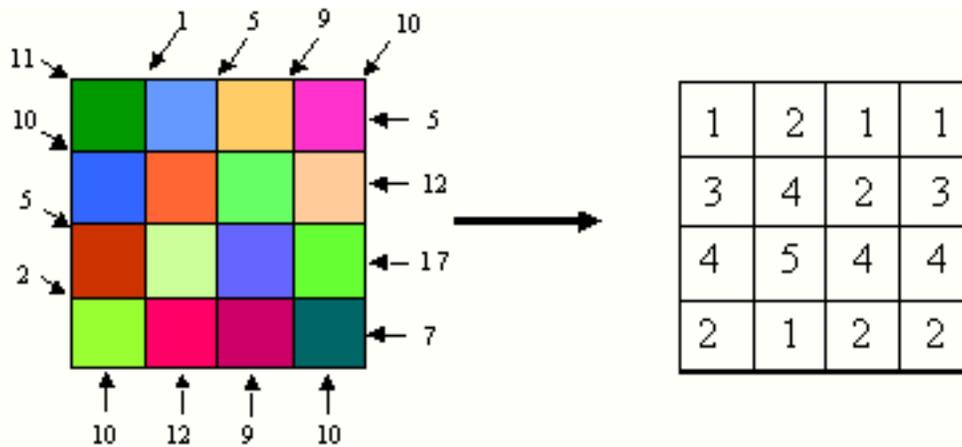


Figure 2-2 : Interpolation des valeurs des voxels dans une coupe TDM.

Dans la Figure 2-3, on montre l'évolution de la disposition de l'émetteur et des détecteurs d'un scanner TDM pour différentes générations.

Pour les appareils TDM de première et deuxième génération, la source et les détecteurs étaient déplacés le long d'un axe lors de l'acquisition d'une coupe, puis, il faut mettre en rotation tous les éléments de ce système afin d'effectuer des mesures de différentes incidences. La translation de ce système est nécessaire pour obtenir la coupe suivante. Les scanners de deuxième génération se différencient de la première par l'utilisation de plusieurs détecteurs.

Les appareils TDM de troisième génération se caractérisent par l'utilisation d'un faisceau du rayon X plus large avec un nombre important de détecteurs, cette évolution permet d'éviter le déplacement de la source comme dans les générations précédentes, cela permet aussi de diminuer le temps d'acquisition.

Dans la quatrième génération de ces appareils, on a des détecteurs sous forme d'un cercle qui sont positionnés autour de la zone à étudier. L'élément qui doit être en rotation avec cette disposition est la source.

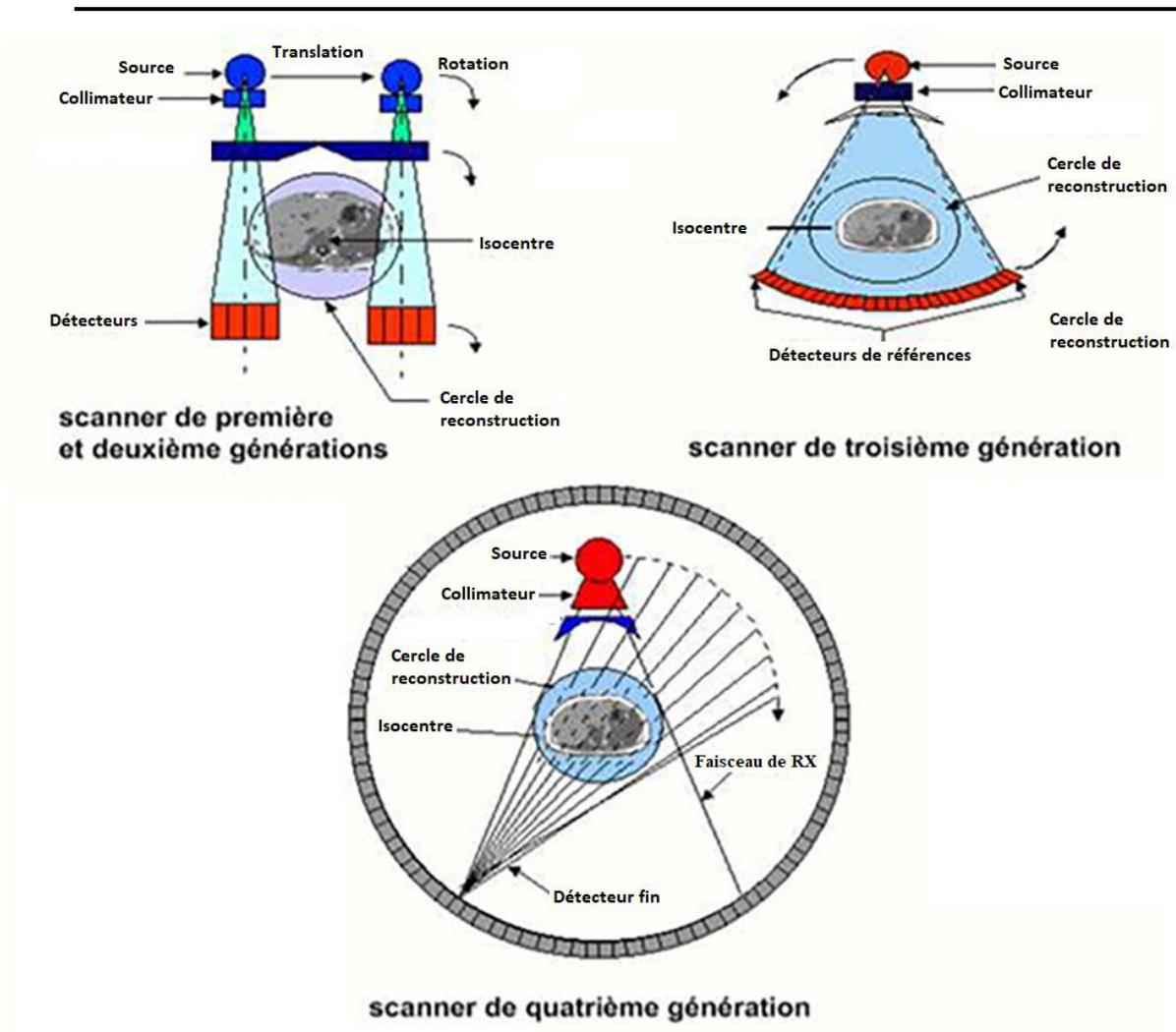


Figure 2-3 : Fonctionnement général des différentes générations de scanners.

À l'heure actuelle, nous disposons du TDM hélicoïdale dans lequel le système émetteur / détecteur est en rotation permanente autour de l'objet, alors que l'objet est en mouvement continu dans la direction perpendiculaire pour acquérir un volume de données complet.

Cette technique permet des analyses rapides, en raison du mouvement de rotation continu du système émetteur / détecteur. Cela permet d'économiser le temps nécessaire pour accélérer et ralentir ces pièces lourdes du scanner. Depuis 1999, plusieurs couches d'émetteurs et de détecteurs sont combinées pour créer plusieurs coupes en même temps.

L'introduction de ce concept exige des algorithmes de reconstruction avancés, puisque les rayons X émis ne sont plus limités à un seul plan. Aujourd'hui, nous disposons des TDMs multi-coupes capables d'acquérir jusqu'à 320 coupes à la fois (voir Figure 2-4).



Figure 2-4 : Une photo d'un scanner moderne (Siemens, Somatom Sensation).

2.2.1.2. Restitution d'une image

Les valeurs d'intensité calculées représentent les densités de l'objet scanné. En imagerie médicale, ces valeurs d'intensité sont normalisées avec les unités Hounsfield (HU). Cette normalisation correspond à la plage de données dans une gamme de 12 bits où l'intensité de l'eau est associée à zéro, et celle de l'air est associée à -1000.

Cette normalisation est formalisée dans la formule ci-dessous, où μ_{H2O} représente la valeur d'intensité de l'eau.

$$\mathbf{HU} = \frac{\mu - \mu_{H2O}}{\mu_{H2O}} * 1000.$$

Tableau 2-1 représente la liste des valeurs de Hounsfield pour certains types de tissus et organes. Le tableau indique que les organes parenchymateux, tels que le cœur et le foie, ont des valeurs d'intensité similaires.

Substance	I_H
Os	50 ↔ 3000
Rein	30 ↔ 50
Cœur	20 ↔ 50
Foie	20 ↔ 60
Pancréas	10 ↔ 40
Eau	0
Tissu adipeux	-220 ↔ -30
Tissu pulmonaire	-900 ↔ -170
Air	-1000

Tableau 2-1 : les valeurs Hounsfield pour différents types de tissus [De (Timothy, 2010)].

Les principaux paramètres concernant la résolution des données TDM sont :

- Le nombre des coupes,
- Le nombre des pixels par coupe,
- Les dimensions de voxels,
- Le pitch
- L'incrément de table

Le nombre de pixels dans une coupe est également appelé matrice d'image. Si nous avons un ensemble de données TDM avec une matrice d'image de 512×512 et 300

coupes, cela correspond à un ensemble de données de volume avec la résolution de $512 \times 512 \times 300$. L'incrément de table indique l'avancement de la table (où l'objet étudié est positionné) à travers le scanner pendant une rotation du portique. Le pitch \mathbf{p} définit le rapport entre l'incrément de table d et la collimation totale des coupes.

La formule utilisée pour calculer \mathbf{p} est : $\mathbf{p} = d / (M \cdot S)$.

- d : l'incrément de table.
- M : représente le nombre des coupes par rotation.
- S : l'épaisseur de coupe.

Par exemple, nous avons un TDM de 16 coupes par rotation ($M = 16$), une épaisseur de coupe de $S = 0,5$ mm, et une incrémentation de table de $d = 6$ mm, le pitch \mathbf{p} est 0.75. Avec un pitch plus grand que 1, nous avons des lacunes dans les données, alors avec un pitch inférieur de 1 indique que les coupes se chevauchent.

L'interprétation correcte des données TDM nécessite une compréhension des artefacts les plus sévères. Un artefact est une différence systématique entre les unités Hounsfield mesurées et les coefficients d'atténuation réels (J.F. Barrett & N. Keat, 2004). En raison du processus de reconstruction complexe, les données TDM peuvent être perturbées à cause des artefacts, ce qui dégrade la qualité de l'image et la rend diagnostiquement inutilisable.

Les artefacts sont dus à :

- Les processus physiques de l'acquisition des données,
- Les facteurs liés au patient,
- Les imperfections dans le scanner (J.F. Barrett & N. Keat, 2004).

Les imperfections dans le scanner doivent, bien sûr, être évitées. Régulièrement, les scanners sont contrôlés et étalonnés en utilisant des spectres de différentes tailles et formes. Malgré les développements actuels sur la réduction ou l'évitement des artefacts, un certain nombre d'objets doivent encore être pris en considération (M.Yazdi & L. Beaulieu., 2008).

QUELQUE EXEMPLE DES ARTEFACTS :

Les artefacts provenant des implants métalliques : Ils se produisent avec les éléments métalliques non démontables dans le scanner, tels que les implants dentaires ou vasculaires.

Sans algorithmes de correction efficaces, ces artefacts entravent l'interprétation dans une grande région de l'image. Les algorithmes de correction ne peuvent pas reconstruire de manière fiable l'interface entre les implants métalliques et leur contour, mais ils permettent de supprimer fortement la surface des effets néfastes.

Étant donné que ce genre d'artefacts est particulièrement sévère (voir Figure 2-5), il a eu une large attention et de nombreux algorithmes de correction qui ont été mis au point à cet effet (Kalender, et al., 1987) (Watzke & W. A. Kalender, 2004).

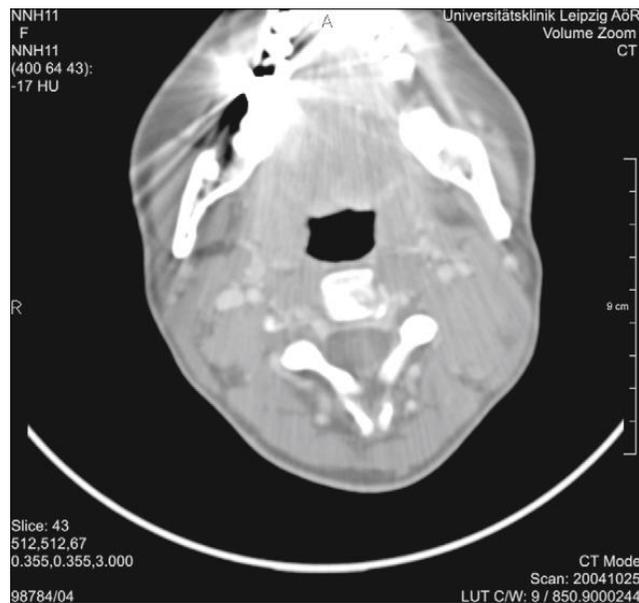


Figure 2-5 : Une coupe de TDM où les implants métalliques dans les dents entravent fortement le diagnostic.

Les artefacts de dispersion : se produisent principalement dans les scanners TDM Multi coupe ayant une couverture large et rapide dans le sens longitudinal. Les artefacts de dispersion sont provoqués par des photons dispersés (Zbijewski & F. J. Beekman, 2006). Artefacts de dispersion font apparaître des ombres foncées entre les os. Diverses

techniques logiciels de correction ont été proposées (Sabo-Napadensky & O. Amir., 2005).

Les artefacts de mouvement : Ces artefacts sont dus au mouvement de l'objet analysé, par exemple, le mouvement du patient, la respiration, le rythme cardiaque ou la relaxation musculaire. Ils apparaissent comme des régions floues où l'anatomie n'est pas fortement représentée. Le flou du mouvement peut être réduit par un certain nombre de techniques. De toute évidence, l'imagerie rapide pendant l'apnée réduit les artefacts de mouvement.

2.2.1.3. Utilisation de la tomодensitométrie

Actuellement, les appareils TDM permettent de faire la différence entre des tissus ayant une densité d'absorption μ qui diffère d'environ 0.3%. Par contre, la valeur du bruit existant dans les images est souvent supérieure à cette valeur. Cela diminue l'efficacité de ces appareils pour différencier entre les tissus du volume étudié, surtout avec l'utilisation des algorithmes qui se basent sur la reconnaissance automatique.

Pour dépasser ce problème, on a la possibilité d'injecter un produit de contraste pour faire varier la valeur d'absorption μ de certains tissus ou organes, ce produit se fixe dans les régions qui nous intéressent pour augmenter sa valeur d'absorption aux rayons X, à fin d'être distingué par rapport au milieu environnant.

Dans la Figure 2-6, on présente une image obtenue par TDM après l'injection d'un produit de contraste, la vascularisation hépatique, les reins et l'aorte peuvent être distingués facilement, contrairement aux images issues du même volume étudié sans injection.



Figure 2-6 : Visualisation d'une image TDM après injection de produit de contraste.

En guise de complément d'information on peut signaler qu'on n'a pas des contre-indications particulières quant à l'utilisation de TDM. Cependant, cette modalité est déconseillée aux patients jeunes et aux femmes enceintes, à cause de l'émission des rayons ionisants.

2.2.2. Imagerie par résonance magnétique

L'imagerie par résonance magnétique (IRM) est basée sur la mesure de la fréquence de résonance (ou spin) des atomes dans des champs magnétiques. Comme la tomodensitométrie, l'IRM est capable de capturer des images en coupe du corps. Par contre, il utilise un rayonnement non-ionisant.

Le contraste de l'image en IRM est déterminé en raison d'un certain nombre de facteurs, y compris l'environnement biochimique des molécules d'eau, de déplacement et de diffusion de liquide et la densité des molécules d'eau dans le tissu. Par conséquent, l'IRM fournit un plus grand contraste entre les tissus mous du corps que la tomodensitométrie. Cela rend l'IRM particulièrement utile en neurologie, oncologie (cancer) et dans l'imagerie musculosquelettique.

Bien que l'IRM fournisse beaucoup de contraste des tissus mous à l'observateur humain, l'intensité de l'image en IRM est très relative et peut même varier pour le même matériau à l'intérieur de la même image. Cela rend plus difficile de faire des mesures quantitatives basées sur l'intensité de l'image. Cela impactera, comme nous allons le voir dans les méthodes de segmentation (voir § 2.3.2).

Dans la section suivante, nous représentons le fonctionnement général d'un appareil IRM, nous y rappelons le principe physique de cette technique, puis la restitution de l'image et finalement son utilisation médicale.

2.2.2.1. Principe Physique de la Résonance Magnétique

Le principe de base derrière l'IRM est le comportement des noyaux atomiques dans un champ magnétique. On peut associer à chaque noyau un spin en fonction du nombre de protons et de neutrons, ce spin se sera nul dans le cas où le nombre de protons et de neutrons sont égaux. Le noyau d'hydrogène est un exemple d'un noyau dont le spin est non nul (possède uniquement un proton).

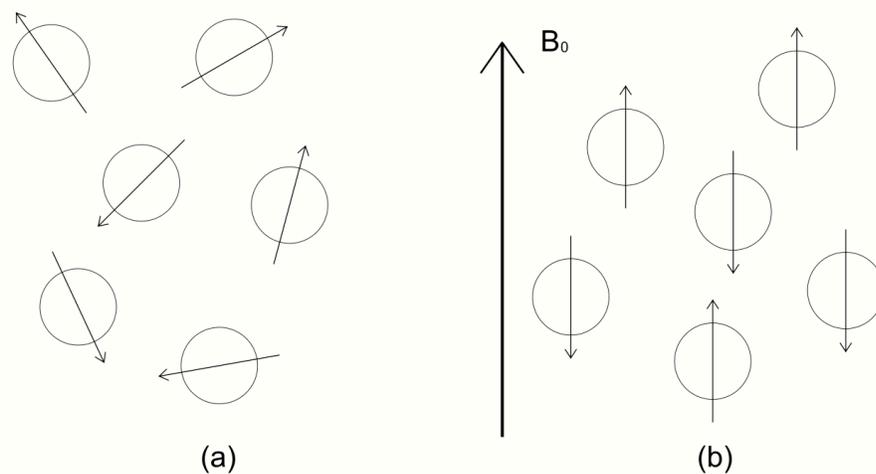


Figure 2-7 : Représentation du spin de particules soumises à un champ magnétique.

Le spin d'un noyau se comporte comme une particule magnétique, ces particules ont une orientation aléatoire, cet état est représenté dans la Figure 2-7 (a).

En appliquant un champ magnétique d'intensité B_0 avec une direction noté \vec{B}_0 , ces particules vont prendre une orientation parallèle ou antiparallèle au champ \vec{B}_0 , cet état est représenté dans la Figure 2-7 (b).

Après l'application de ce champ magnétique, on aura une très petite quantité de ces particules qui s'orientent en préférence dans la même direction de \vec{B}_0 . Cela se traduit par une très petite augmentation du champ \vec{B}_0 , ce champ magnétique supplémentaire est noté \vec{M} .

Puis, on applique une impulsion électromagnétique (noté \vec{B}_1), la fréquence de cette impulsion est déterminée selon la fréquence de résonance de la particule, le but de l'application de \vec{B}_1 est de faire basculer le vecteur \vec{M} dans le plan normal au vecteur \vec{B}_0 .

La Figure 2-8 montre le changement de l'orientation du vecteur \vec{M} après l'activation de l'impulsion électromagnétique \vec{B}_1 , le vecteur \vec{M} devient sur un plan orthogonal à \vec{B}_0 , en tournant autour de ce dernier.

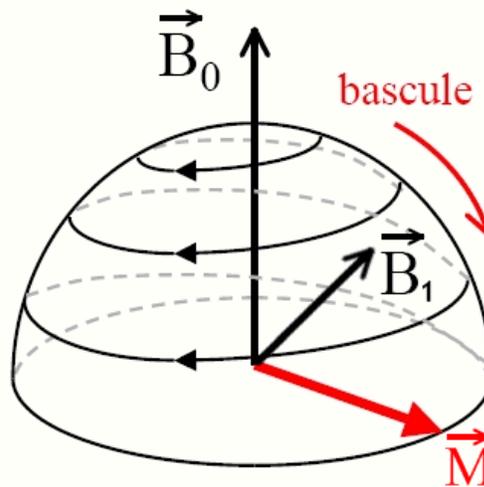


Figure 2-8 : Effet d'une impulsion électromagnétique.

A la fin de l'impulsion électromagnétique, on aura le phénomène de relaxation, le vecteur \vec{M} revient à sa valeur initiale (même direction et sens de \vec{B}_0), tout en gardant le mouvement de rotation autour de l'axe de \vec{B}_0 .

La Figure 2-9 illustre le comportement du vecteur \vec{M} dès la fin de l'impulsion, il va redevenir colinéaire à \vec{B}_0 d'une manière progressive.

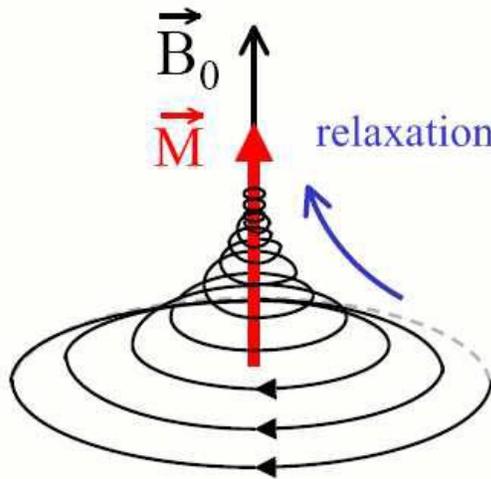


Figure 2-9 : Relaxation après impulsion.

A partir du phénomène de relaxation, on peut avoir deux mesures temporelles :

- T1 : le temps de relaxation longitudinale.
- T2 : le temps de relaxation transversale.

La relaxation de ces deux mesures ne se fait pas de la même manière, le temps nécessaire à la relaxation de T1 est toujours supérieur à celui de T2.

2.2.2.2. Construction d'une image en Résonance Magnétique

La construction d'une image IRM se base essentiellement sur la mesure de la relaxation T1 ou T2 sur les différentes régions du volume étudié. C'est pour cela qu'un

dispositif d'imagerie IRM est toujours constitué d'un gros aimant, des émetteurs radio qui délivrent des impulsions, une antenne de réception de la radio et de l'électronique qui décodent le signal.

Afin de localiser en trois dimensions l'origine du signal réémis par les particules soumises au champ \vec{B}_1 , on applique un gradient de champ magnétique \vec{B}_0 , de sorte que l'intensité du champ magnétique diminue de façon linéaire selon un certain axe, la fréquence de résonance des atomes dépendra de leur position selon l'axe.

Si nous imaginons le simple exemple de deux tissus physiquement séparés qui subissent le même champ magnétique, nous aurions seulement un «pic» dans le spectre de résonance magnétique. Mais en appliquant un gradient magnétique de telle sorte que les deux tissus subissent différentes intensités du champ magnétique, chaque tissu aura une fréquence de résonance différente et on pourrait identifier quelles sont les parties du signal qui proviennent d'un tissu. Ceci est en fait une simplification. Dans la pratique, l'acquisition IRM réel implique des applications complexes, il faut bien choisir les fréquences des impulsions, les gradients des champs magnétiques, et les mesures de fréquence radio.

Dans la Figure 2-10, nous montrons le gradient de champ magnétique appliqué dans les trois directions dans l'IRM : gauche/droite, devant/derrière et haut/bas. Ces gradients sont générés à l'aide des bobines disposées d'une manière parallèle aux trois axes.

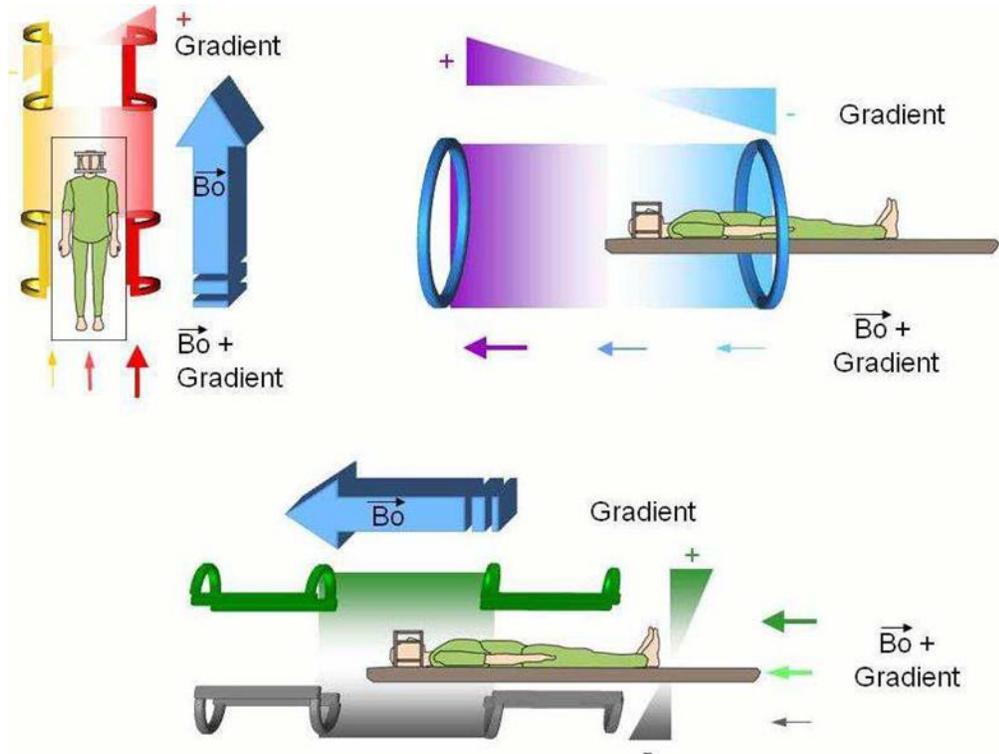


Figure 2-10 : Gradient magnétique généré dans les trois axes de l'espace.

Le résultat obtenu après la restitution est lié généralement à deux variables :

- TR : temps de répétition entre deux impulsions.
- TE : temps entre l'impulsion et la réception de l'écho.

En changeant ces deux paramètres, on peut avoir un signal qui possède des propriétés différentes.

En utilisant un TR et TE courts, le contraste de l'image sera basé sur la lecture du signal longitudinale, donc l'image obtenue sera pondérée en T1. Cette pondération, par exemple, permet de différencier entre la substance blanche et la substance grise dans le cerveau.

En utilisant un TR et TE longs, le contraste de l'image est basé sur la lecture du signal transversal, donc l'image sera pondérée en T2. Cette pondération permet, par exemple, de supprimer le signal de l'eau libre dans une image.

En utilisant un TR long et un TE court, le contraste de l'image obtenue sera équivalent aux densités protoniques du volume étudié. La Figure 2-11 montre les images obtenues de la même coupe avec deux pondérations T1 et T2 différentes. La première pondération permet de mettre en valeur un œdème et la deuxième pondération permet d'avoir une image moins bruitée, ce qui aide à déterminer l'anatomie du cerveau.

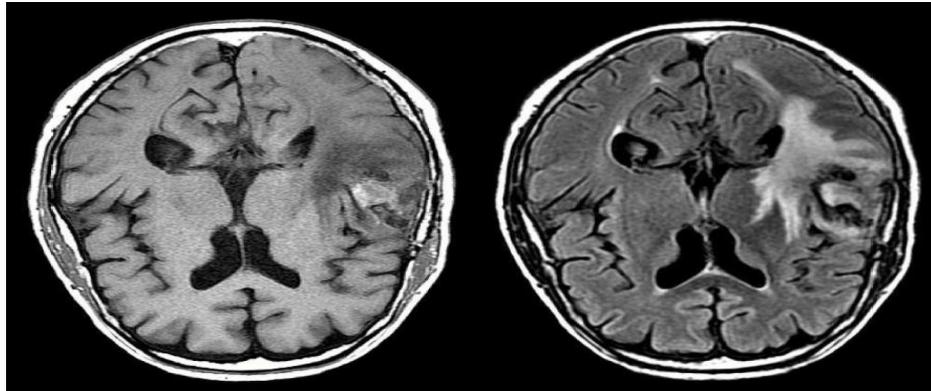


Figure 2-11 : IRM cérébrale en pondérations T1, FLAIR.

2.2.2.3. Utilisation de l'Imagerie par Résonance Magnétique

Les appareils IRM actuels utilisent une intensité du champ magnétique entre 0.1 et 7 teslas, sachant que 1 tesla équivaut à 20 000 fois le champ magnétique terrestre. La valeur de l'intensité du champ magnétique permet de déterminer la résolution de l'image obtenue. A titre d'exemple, on obtient une résolution de l'ordre du millimètre avec un IRM de trois teslas.

L'IRM permet d'obtenir des images meilleures par rapport à la tomodensitométrie au niveau du contraste entre les tissus mous du corps. L'IRM est largement utilisée dans les examens neurologiques, oncologiques (cancer), l'imagerie musculosquelettique et crânienne. C'est une méthode qui utilise le rayonnement non-ionisant. L'énergie émise par cette technique n'entraînant pas de modification au niveau des molécules et des atomes, son utilisation ne présente donc pas de dangers pour la santé des patients. Cependant, dans la pratique, l'utilisation du TDM est beaucoup plus importante que l'IRM, et cela même dans le cas où les deux modalités sont égales au niveau de l'efficacité

du diagnostic. Cela est dû essentiellement au coût beaucoup plus cher de l'IRM par rapport au TDM.

Les artéfacts qui peuvent diminuer l'efficacité du diagnostic d'une image IRM sont essentiellement dus aux mouvements du patient ou de la perturbation du champ magnétique à cause d'un objet métallique (plombage, pacemaker, prothèse).

La mise au point de méthodes d'imagerie par IRM est encore un domaine de recherche très actif. Par exemple, on peut détecter avec cette modalité l'activité du cerveau (IRM fonctionnel) : En abaissant la résolution et en effectuant de nombreux balayages au fil du temps, des taux variables de l'oxydation peuvent être détectés dans le cerveau. Cela permet de faire des mesures fonctionnelles de l'activité cérébrale. Des chercheurs ont même réussi à reconstruire des images rudimentaires de ce qu'un patient regarde (Miyawaki, et al., 2008) : Après l'affichage d'une série d'images qui contient des lettres, les chercheurs ont été en mesure de générer des images d'une résolution très faible des lettres que le patient voit uniquement à partir de l'analyse de son cerveau. Ce résultat est obtenu en utilisant des séquences d'impulsions d'IRM personnalisées et certaines applications de l'apprentissage automatique (voir Figure 2-12).

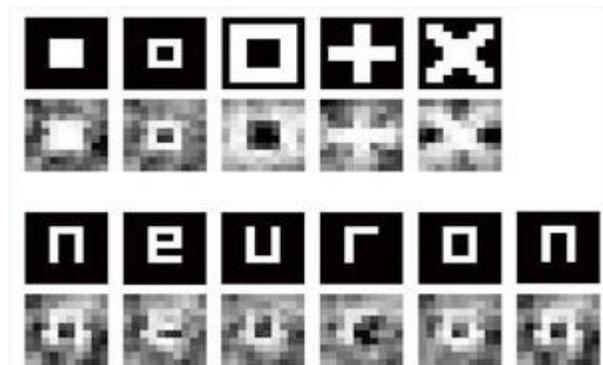


Figure 2-12 : Des images reconstruites à partir du cerveau (IRMf) Tout en montrant les lettres correspondantes (Miyawaki, et al., 2008).

Note: Les chercheurs ont utilisé des corrélations apprises entre les régions du cerveau activées tout en montrant au patient un ensemble d'images d'entraînement aléatoires.

2.2.3. Imagerie en médecine nucléaire

Dans cette section, nous décrivons la Tomographie par émission de positons (TEP) et la tomographie par émission mono photonique (TEMP). Ces deux modalités qui sont essentielles en médecine nucléaire ainsi que leur combinaison avec les modalités décrites précédemment TDM et IRM, conduisant à des scanners hybrides, tels que le TEP / TDM.

2.2.3.1. Tomographie par émission de positons

Tomographie par émission de positons (TEP) est une technique d'imagerie en médecine nucléaire. La TEP génère des images 3D de processus fonctionnels, tels que l'activité métabolique. Une substance radiopharmaceutique de courte durée 'traceur' est injectée à travers le corps jusqu'à la zone cible. La substance est traitée par le métabolisme et les désintégrations isotopes radioactifs conduisent à l'émission des positons.

L'interaction d'un positron avec un électron génère deux photons gamma (rayons) dans des directions opposées (180°). Ces photons sont ensuite mesurés par l'ensemble de détecteurs du scanner TEP (voir Figure 2-13). Puisque les photons de cette interaction sont toujours émis dans des directions diamétralement opposées, il est possible de localiser leur source au long d'une ligne droite. La reconstruction de l'image est très similaire à la reconstruction des données TDM. Puisque les traceurs sont traités par le métabolisme, la détection de l'emplacement d'une activité métabolique (par exemple l'activité cérébrale lors d'une tâche spécifique) devient possible. La résolution spatiale d'une image TEP est nettement plus faible que les données TDM et IRM ; la taille du pixel est égale à 3-4 mm.

Grace à la capacité de caractériser l'activité métabolique, les appareils de TEP jouent un rôle très important dans le diagnostic du cancer et dans la recherche de métastases.

Un exemple particulier et important est la recherche de la tumeur primitive inconnue. La TEP est également utilisée pour évaluer le succès d'une thérapie anticancéreuse (par exemple une chimiothérapie). Souvent, la TEP détermine le succès ou l'échec d'une thérapie antérieure par rapport à TDM ou IRM (H. Young, et al., 1999).

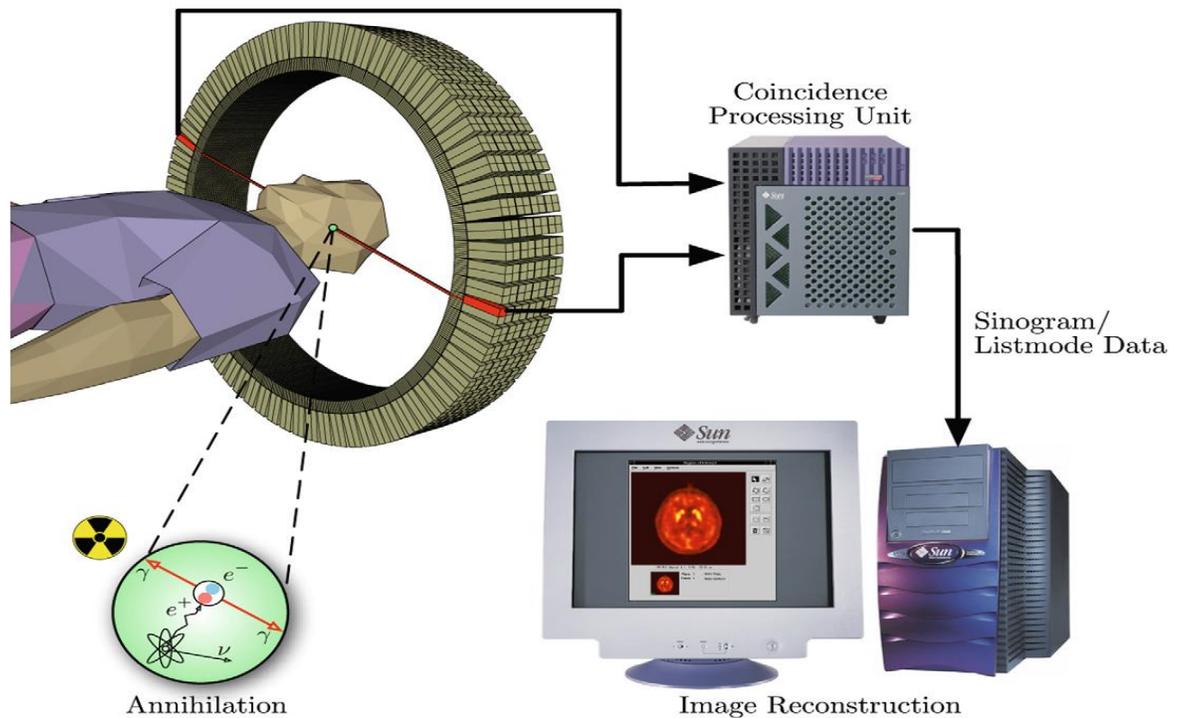


Figure 2-13 : La détection de l'annihilation en TEP.

Note : Dans le processus d'annihilation, deux photons sont émis dans des directions opposées et enregistrés par le TEP dès qu'ils arrivent à l'anneau de détection. Après cela, les données sont transmises à une unité de traitement qui détermine si les deux événements enregistrés représentent un événement de coïncidence. Toutes les coïncidences sont traitées par l'unité de traitement d'image où les données de l'image finale sont produites.

En outre, le TEP est également essentiel en cardiologie (cartographie de la fonction cardiaque et contribution au diagnostic de l'athérosclérose) et en neurologie (représentation de l'activité du cerveau et donc de contribution au diagnostic des maladies neurodégénératives). Pour fournir des informations véritablement intégrées de l'activité métabolique et de l'anatomie, les scanners hybrides ont été développés. Ils intègrent les deux modalités et donc permettent de réduire fortement le problème de recalage, puisque la position du patient est identiquement la même pour les deux procédures d'acquisition (voir Figure 2-14).

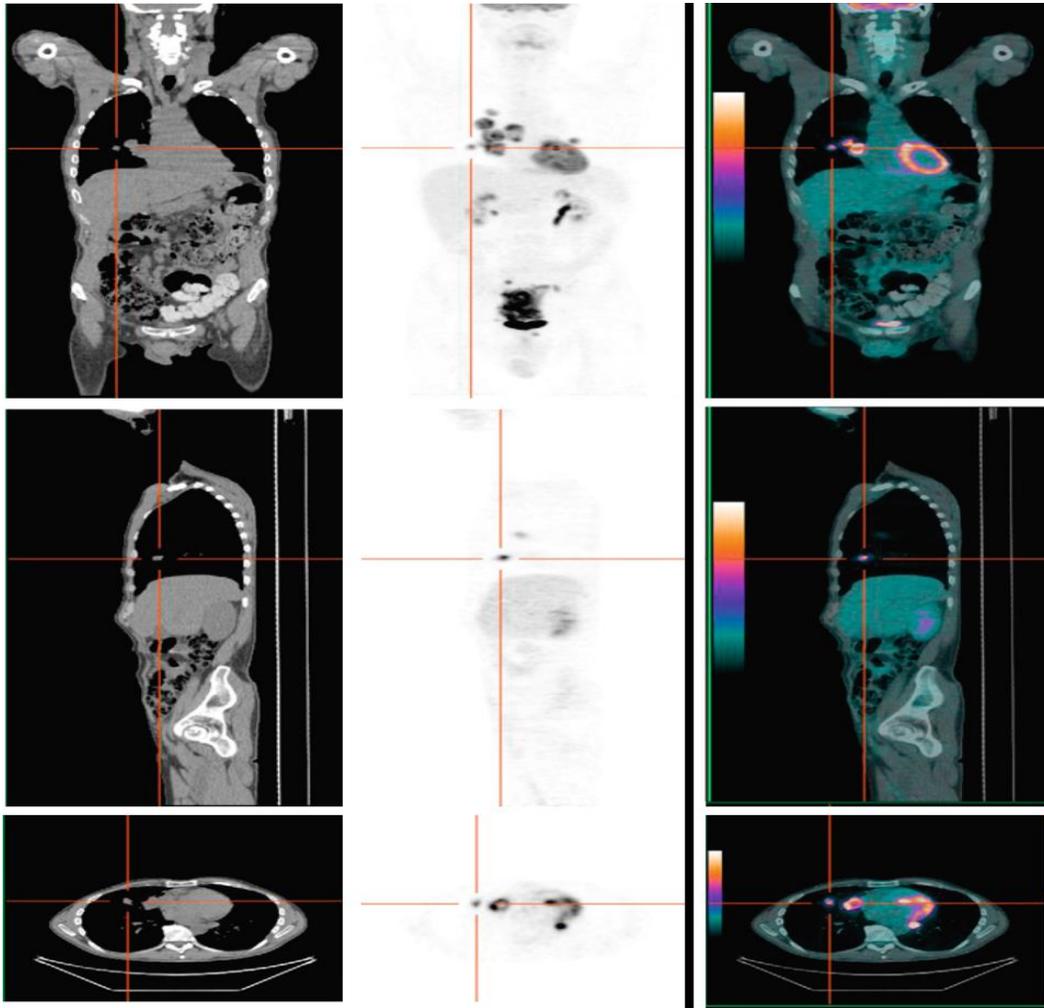


Figure 2-14 : La représentation des données TDM et TEP : isolément (à gauche) et superposées (à droite).

Note: Le TEP / TDM d'un examen de dépistage du carcinome du côlon révèle plusieurs lésions en dehors de la tumeur primaire. Sur la position actuelle du curseur (voir le curseur en croix), un nodule de poumon est montré. Les images ont été acquises avec un GE PET / CT Discovery 600.

2.2.3.2. Tomographie par émission mono photonique

La tomographie par émission mono photonique (TEMP) est une technique qui permet de réaliser des images et des reconstructions en trois dimensions des organes et

de leur métabolisme au moyen d'un ensemble de caméras, sensibles aux rayons Gamma, qui tournent autour du patient.

En général, les projections sont acquises tous les 3 ou 6 degrés avec une rotation complète de 360 degrés. Le temps de balayage total est généralement entre 15 et 20 minutes. Les métaux lourds sont utilisés comme éléments traceurs. Le technétium-99m est le traceur le plus fréquemment utilisé. Il a une faible demi-période et sa radioactivité est faible. Pour l'utilisation en imagerie médicale, le métal lourd doit être combiné avec une substance biologiquement neutre.

Contrairement à TEP, TEMP est plus limitée en termes de résolution spatiale et temporelle, et aussi en ce qui concerne les effets qui peuvent être surveillés. Cependant, les traceurs TEMP se désintègrent plus lentement que les traceurs TEP, permettant ainsi de faire des examens sur des fonctions métaboliques plus durables. TEMP est appliquée, par exemple, pour le diagnostic en oncologie, l'imagerie osseuse, dans la neuro-imagerie, et pour étudier la perfusion du myocarde dans le ventricule gauche.

Avec une imagerie TEMP, les photons gamma sont émis à partir des noyaux du traceur. Ces photons doivent être détectés par une caméra gamma. Des collimateurs au niveau de ces caméras sont nécessaires pour estimer leurs chemins. Généralement, l'espacement des voxels est compris entre 4 et 7 mm avec des coupes qui se compose de 64×64 ou 96×96 pixels. Les images sont fournies soit sous forme d'images en niveaux de gris ou des images de pseudo-couleur. Comme la TEP, l'imagerie TEMP a été intégrée à l'imagerie TDM dans un scanner hybride (voir Figure 2-15).



Figure 2-15 : Hybrid scanner TEMP / TDM, GE NM / CT 670 de General Electric.

Note : Il contient un scanner TDM de 16 coupes qui peut être utilisé pour toutes les tâches de diagnostic, avec un mode de fonctionnement à faible dose. Le scanner TDM est installé à l'arrière, les deux éléments de détection dans la partie frontale appartiennent au scanner TEMP.

2.3. Traitement et analyse des images médicales

Comme mentionné ci-dessus, les différents traitements d'acquisition en imagerie médicale et l'analyse médicale proprement dite, sont devenus étroitement liés... à tel point qu'il est difficile de discuter de l'un sans l'autre.

Aujourd'hui, l'imagerie médicale connaît, comme les autres industries de l'informatique, une explosion de masses de données. Comme la technologie évolue, les appareils d'imagerie créent de façon exponentielle plus d'informations, et ils deviennent en même temps moins coûteux et plus disponibles, ce qui fait que, de plus en plus de patients profitent de ces nouvelles techniques.

Cette explosion de l'information change radicalement la façon dont les médecins analysent ces données. Ils ont été obligés de contrôler toutes les coupes 2D pour analyser une structure 3D. Actuellement, l'utilisation de nombreuses techniques assistées par ordinateur devient une pratique courante pour analyser ces coupes 2D.

Les questions les plus courantes pour lesquelles les cliniciens souhaitent obtenir des réponses lorsqu'ils regardent une, ou un ensemble d'images, comprennent :

- Quelle est la taille d'une structure particulière ?
- Comment les structures sont-elles liées dans l'espace 3D, qu'elles sont les distances et les relations entre des structures voisines ?
- Comment les données d'une image sont-elles en corrélation avec les données des autres images ?

Les tentatives de réponse à ces questions ont conduit historiquement à trois groupes interdépendants de traitements sur les images médicales. Chaque groupe de ces traitements ayant été l'objet de plusieurs algorithmes et techniques. Ces groupes de traitement sont le recalage, la segmentation et la visualisation. Nous discutons brièvement de chacun d'eux dans ce qui suit.

2.3.1. Recalage

Le recalage d'image est le processus de transformation de différents types de données en un seul système de coordonnées (Zhang, 1993). Le recalage est nécessaire afin d'être en mesure de comparer ou d'intégrer les données obtenues à partir de différentes mesures. C'est le cas en particulier où les images sont prises à des moments différents, pour différents patients, ou en utilisant différentes modalités d'imagerie. Après le recalage d'image, les emplacements anatomiques corrélés de différentes images seront alignés.

Les informations collectées au fil du temps, à partir de différents patients, ou différentes modalités sont souvent complémentaires. Pour un seul patient, un médecin pourrait vouloir aligner les informations fonctionnelles à partir d'un examen TEP avec les détails structurels fournis à partir d'un TDM, ou les changements structurels entre deux examens de même organe. Pour plusieurs patients, un médecin pourrait vouloir recueillir une carte statistique de la fréquence de certains cancers qui se trouvent dans le cerveau. Le recalage de l'image permet la fusion automatique ou semi-automatique de plusieurs

ensembles d'examens dans un espace commun, ce qui représente un outil de diagnostic beaucoup plus puissant que l'analyse individuel des images.

Le recalage d'images d'un seul patient peut parfois être accompli grâce à l'utilisation des marqueurs de référence, qui sont fixés de façon rigide chez le patient, ou grâce à l'utilisation manuelle de marqueurs anatomiques (par exemple le bout du nez). Dans ce cas, le recalage devient un processus assez simple pour aligner exactement un ensemble de correspondances de points connus afin de pouvoir apprécier les autres différences entre ces images.

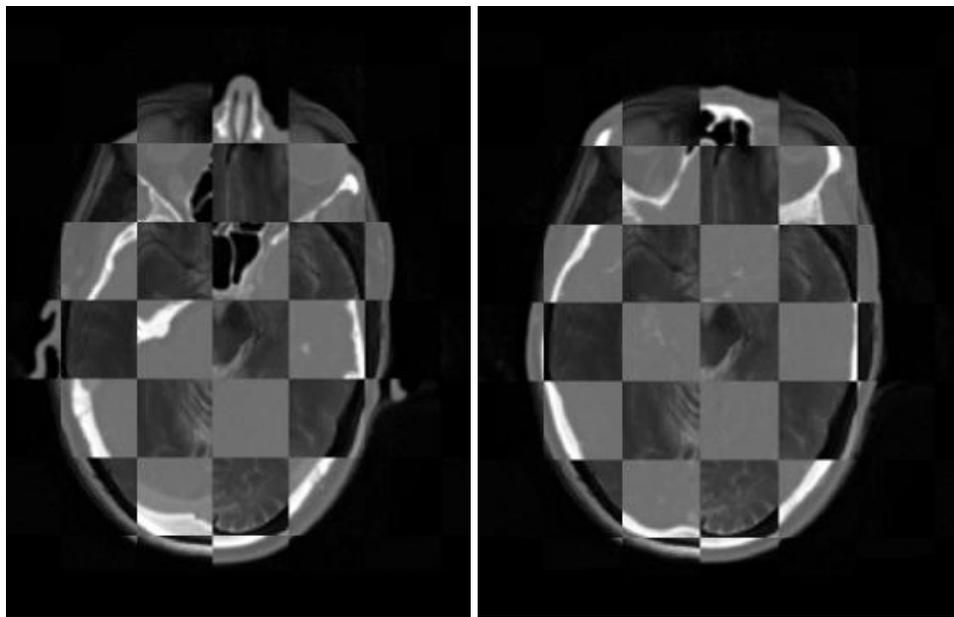


Figure 2-16 : Illustration de recalage de deux modalités complémentaires.

Note : Le motif en damier est utilisé pour montrer le décalage (à gauche) et l'alignement après le recalage (à droite) (Sonny, 2007).

Toutefois, lorsque les marqueurs de référence ne sont pas disponibles, le recalage devient une tâche algorithmiquement plus difficile. Au sein d'une seule modalité, la détection automatique de certaines caractéristiques peut être utilisée comme marqueurs, suivie par l'alignement rigide (sans modification de la taille de l'image) ou non rigide de

ces caractéristiques adaptées. Pour différentes modalités, les caractéristiques d'une image peuvent ne pas correspondre aux mêmes caractéristiques de l'autre, ce qui rend la comparaison encore plus compliquée.

La méthode de référence pour l'alignement de différentes modalités utilise une mesure d'erreur qui calcule l'alignement entre deux images. Pour aligner les images, un algorithme d'optimisation numérique est utilisé. Il devine itérativement la bonne transformation, évalue l'indicateur d'erreur (information mutuelle) et compare à nouveau cette erreur devinée avec celle de la transformation suivante.

Autrement dit, étant donné une image de référence (par exemple, un scanner du cerveau), et une seconde image qui doit être mise dans le même système de coordonnées, cette deuxième image est déformée jusqu'à ce que la correspondance de l'information mutuelle entre elles soit maximisée. En raison de l'incertitude liée à ce traitement, un médecin sera généralement impliqué pour vérifier le résultat obtenu.

2.3.2. Segmentation

Dans le cadre de cette thèse, nous sommes principalement concernés par la segmentation qui est une étape préalable à la visualisation. La segmentation consiste à décomposer les données d'une image source relativement à des structures significatives qui sont pertinentes pour une tâche médicale spécifique.

La segmentation a deux aspects pour les objets concernés par ce traitement :

- Ils doivent être identifiés comme des structures anatomiques particulières.
- Ils doivent être délimités dans la mesure où leurs frontières sont précisément spécifiées.

La reconnaissance est une tâche de haut niveau que les humains effectuent habituellement mieux que les ordinateurs. D'autre part, la délimitation est une tâche qui demande beaucoup de précision. Des algorithmes évolués peuvent conduire à d'excellentes solutions pour ce problème. Un défi pour les processus de segmentation consiste donc à combiner la force de l'utilisateur avec le potentiel d'assistance informatique.

Techniquement, une étiquette unique est attribuée à chaque voxel représentant l'appartenance à une structure particulière. Dans les applications médicales, il est généralement suffisant de segmenter seulement la structure concernée et non pas toutes les structures anatomiques. D'autre part, la segmentation peut parfois être appliquée pour supprimer certaine structure qui empêche la visualisation. A titre d'exemple, les os dans les données TDM d'angiographie sont souvent segmentés pour être retirés de la visualisation des structures vasculaires (Fiebich, et al., 1999).

2.3.2.1. Exigences de la segmentation

Avant de décrire les stratégies de segmentation, il est important de rappeler les exigences essentielles de ce processus :

- **Robustesse** : Pour être utile dans la routine clinique, les solutions d'analyse d'image doivent être robustes. Elles doivent fonctionner pour une grande variété de cas. Il est acceptable que l'effort (humain) d'interaction augmente dans les cas difficiles avec des anomalies dans l'anatomie ou dans les structures pathologiques, mais il devrait avoir un soutien même dans ces cas. Les solutions automatiques qui fonctionnent seulement dans certaines circonstances pourraient être intégrées dans un logiciel, mais doivent être complétées par une autre solution (manuelle par exemple).
- **Précision** : la segmentation de l'image et les opérations ultérieures doivent fournir des résultats exacts. Les études de validation sont nécessaires pour déterminer que la précision obtenue est acceptable.
- **Reproductibilité** : Un seul utilisateur obtiendrait un résultat similaire si l'algorithme est appliqué dans plusieurs reprises (reproductibilité intra-observateur) et les différents utilisateurs pourraient également produire des résultats similaires (reproductibilité inter-observateur). Cette exigence est plus facile à prouver, car aucune corrélation sur le terrain n'est nécessaire.
- **Vitesse** : on pourra avoir d'excellentes solutions mais qui sont informatiquement extrêmement exigeantes. Ces solutions ne seront pas utiles si on n'arrive pas à les accélérer à un niveau d'interactivité acceptable.

La puissance croissante des ordinateurs allège cette situation. Cependant, les tailles croissantes des données d'image aplatissent souvent la puissance de calcul supplémentaire.

Ces exigences sont générales, l'importance de chacune d'elle diffère fortement par rapport au contexte d'application. La vitesse peut être l'exigence dominante si l'analyse d'images est effectuée sous une forte pression du temps, par exemple dans la salle d'opération. Reproductibilité est cruciale, si les données radiologiques sont analysées pour surveiller la progression d'une maladie (des examens de suivi). Cependant, la précision est essentielle pour la segmentation de la tumeur, ainsi que pour la segmentation des structures à risque. Des discussions similaires des exigences peuvent être trouvées dans (Hahn., 2005) et (Udupa & Herman, 2000).

2.3.2.2. Segmentation manuelle

La méthode la plus générale et la plus facile pour la segmentation d'images est le dessin manuel dans les coupes des données radiologiques. L'utilisateur décrit les structures pertinentes avec un dispositif de pointage. Pour modifier ce contour, il est souvent possible de redessiner certaines parties pour remplacer d'autres parties précédemment dessinées. Bien que la souris soit fréquemment utilisée pour la segmentation manuelle, d'autres dispositifs de pointage, comme une tablette graphique, peut mieux servir dans ces tâches.

La segmentation manuelle est toujours applicable. Cependant, il est fastidieux de ne pas pouvoir assurer les exigences de reproductibilité et de précision, car l'utilisateur s'écarte souvent –même si légèrement- par rapport au contour souhaité. Malgré ces problèmes, la segmentation manuelle est très répandue, en particulier, si les objets sont très difficiles à définir en raison d'artefacts, de faible contraste, ou une forme inattendue.

A titre d'exemple, la segmentation de la tumeur est souvent réalisée manuellement, car la biologie tumorale variable empêche le soutien automatique dans de nombreux cas.

2.3.2.3. Segmentation basée sur un seuil

La génération d'une image binaire est basée sur la sélection d'un seuil global ou avec un intervalle (unique ou multiple). Tout type de segmentation à seuil est basé sur l'analyse de l'histogramme de l'image.

Habituellement, cela ne produira pas une segmentation de la structure cible immédiatement, mais il peut servir comme point de départ. De nombreux types de segmentation utilisent la segmentation à seuil soit comme une première étape, ou pour affiner les résultats obtenus avec d'autres algorithmes.

Une application typique de la segmentation à seuil est l'identification des os dans les données TDM, qui peuvent être caractérisées par des valeurs de Hounsfield. Cependant, même dans cette application, la précision de la segmentation à seuil est limitée, car les os minces généralement ne sont pas correctement identifiés.

2.3.2.4. Région croissance

La région croissance est une famille d'algorithmes de segmentation qui permet de déterminer des régions volumétriques connectées. Le processus de croissance est initié par un ou plusieurs points de départ. Ces points sont dans la plupart des cas définis par l'utilisateur, mais ils peuvent également être le résultat d'une analyse automatique.

Le processus de croissance assemble successivement les voxels voisins jusqu'à ce qu'un critère d'inclusion sélectionné par l'utilisateur ne soit plus rempli (voir Figure 2-17). Ce critère d'inclusion est généralement un seuil pour les valeurs d'intensité (collecter tous les voxels avec une intensité inférieure ou égale à l'intensité des points de départ et au-dessus du seuil). Avec cette approche, l'utilisateur indique simplement les points de départ qui appartiennent à la structure cible. En outre, l'utilisateur doit fournir le seuil comme paramètre supplémentaire. La sélection de seuil est souvent un processus d'essais et d'erreurs : la segmentation est démarrée avec un certain seuil qui sera modifié jusqu'à ce que le résultat souhaité soit atteint.

La région croissance de base repose sur une condition fixe, habituellement un ou deux seuils. Pour l'ajustement interactif de ces paramètres, la stratégie de base est trop lente. (Boiselle, et al., 2002) a introduit une région progressive croissante dans laquelle

la segmentation est effectuée pour un ensemble de seuils en parallèle. Cela peut être efficacement accompli en utilisant le fait que le jeu de voxel segmenté pour un seuil plus élevé est complètement inclus dans l'ensemble de voxels pour un seuil inférieur.

Région croissance est souvent utilisé pour la segmentation des structures vasculaires avec contraste rehaussé (Boiselle, et al., 2002) (Boskamp, et al., 2004). Le point de départ placé à la racine de l'arbre vasculaire est suffisant pour délimiter la structure cible.

Dans la pratique, souvent plusieurs points de départ doivent être choisis et les résultats de segmentations relatives à chaque point de départ sont combinés. Après prétraitement approprié, les résultats sont souvent acceptables.

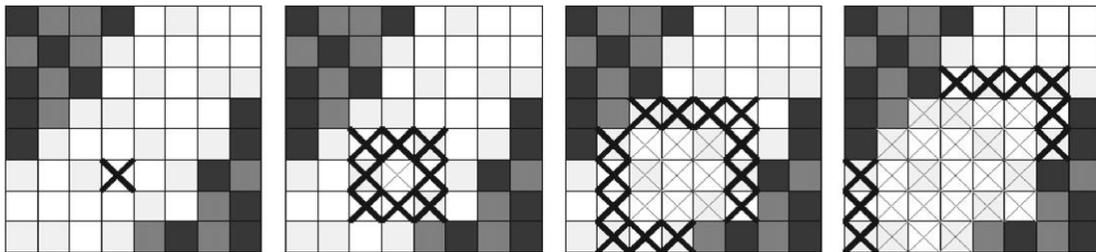


Figure 2-17 : Les différentes étapes d'un algorithme de région croissance.

Note : A partir d'un point de départ défini par l'utilisateur (à gauche), successivement plus des voxels (qui remplissent un critère d'inclusion) sont agrégés, X marque les voxels actifs courants (Boiselle, et al., 2002).

2.3.2.5. Segmentation de Watershed

La segmentation de Watershed est une autre méthode ayant des origines dans la morphologie mathématique (Serra., 1982). Le concept général a été introduit par (Digabel & Lantuejoul., 1978). Une innovation majeure de cette méthode a été réalisée par (Vincent & Soille, 1991), (voir aussi (Hahn., 2005)) pour une discussion sur l'histoire de

la segmentation de Watershed). Depuis cette date, il a été largement appliqué à différentes tâches de segmentation d'images médicales.

Dans la segmentation de Watershed une image est considérée comme un paysage topographique avec des crêtes et des vallées. Les valeurs d'élévation du paysage sont généralement définies par l'intensité des pixels ou de leur amplitude de gradient. En se basant sur une telle représentation 3D, La transformation Watershed décompose une image en bassins versants. Pour chaque minimum local, un bassin versant comprend tous les points dont la trajectoire d'ascendance la plus forte se termine à ce minimum (voir Figure 2-18).

La transformation Watershed décompose complètement une image et donc attribue chaque pixel soit à une région ou à un bassin versant. Avec des données d'images médicales bruyantes, on obtient un grand nombre de petites régions. Ceci est connu comme le problème de sur-segmentation (voir Figure 2-19). Généralement l'image gradient est utilisée comme base pour la transformation Watershed. Cependant, Gradient est fortement sensible au bruit de l'image. Par conséquent, le filtrage approprié est essentiel.

Le problème de sur-segmentation peut être résolu par certains critères pour la fusion des régions. L'utilisateur doit disposer de certaines installations pour influencer le processus de fusion (Hahn & Peitgen., 2003).

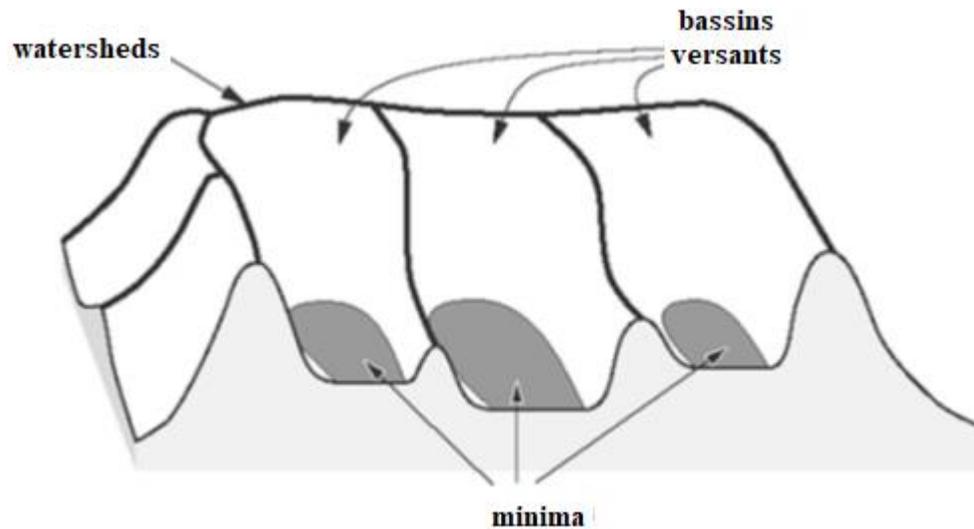


Figure 2-18 : Principe de la transformation du WaterShed où les valeurs d'intensité définissent les collines et les bassins (Hahn., 2005).

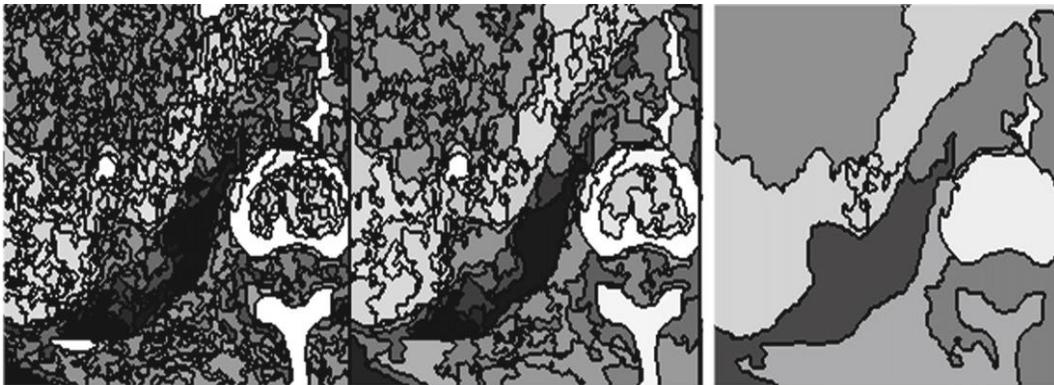


Figure 2-19 : Illustration du problème de sur-segmentation de la transformation Watershed appliqué à une coupe axiale d'une image TDM.

Watershed à base de marqueurs :

L'utilisateur peut spécifier des emplacements dans les images appartenant à la structure cible ou qui n'appartiennent pas à cette structure. Si l'utilisateur spécifie un point d'inclusion et un point d'exclusion, un bassin versant supplémentaire est construit au niveau maximal entre eux (voir Figure 2-20). L'arbre de fusion est parcouru de sorte que chaque région contienne soit des points d'inclusion soit des points d'exclusion, mais pas les deux. Ce style d'interaction est appelé segmentation *Watershed* à base de marqueurs.

Il existe plusieurs variantes de la transformation **Watershed**. Par exemple, la fusion peut prendre en considération les informations de gradient ou d'autres critères d'homogénéité. Une variante fréquemment utilisée pour fusionner les régions, c'est de comparer la différence de la valeur moyenne de niveau de gris à un seuil. Ce processus peut être effectué de manière itérative et se traduit également par un arbre de fusion hiérarchique.

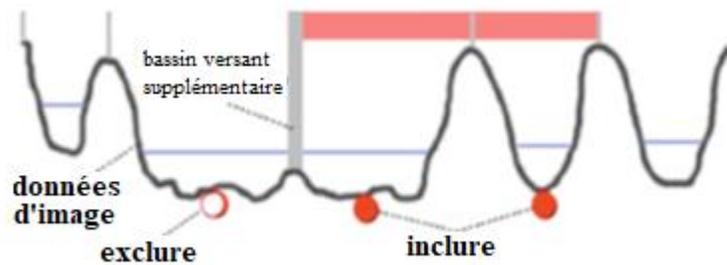


Figure 2-20 : L'utilisation des marqueurs avec Watershed.

Note : Les données d'image sont considérées comme un paysage. Ensuite on inonde progressivement ce paysage obtenant ainsi une succession de petits bassins inondés. Pour empêcher la fusion des bassins intra et extra zone d'intérêt, un watershed est ajouté entre un point d'inclusion et un point d'exclusion spécifié par l'utilisateur.

2.3.3. Visualisation

La visualisation se réfère à l'utilisation des images, des diagrammes et des animations pour communiquer un message. La visualisation médicale est un type spécialisé de visualisation scientifique qui tente de transformer, sélectionner ou représenter des données provenant des simulations ou des expériences pour permettre l'exploration, l'analyse et l'amélioration de la compréhension de ces données (Kersten-Oertel, et al., 2013). Dans le contexte de l'imagerie médicale, la visualisation vise la transformation des données brutes acquises vers une forme plus familière et compréhensible pour le médecin.

Comme exemple concret, considérons les données acquises dans la modalité d'imagerie appelée tenseur de diffusion (DTI). Dans DTI, un ensemble spécialisé de

séquences des impulsions en IRM sont utilisées pour acquérir non seulement une valeur par voxel, mais plusieurs valeurs représentant le tenseur de diffusion du voxel. Le tenseur de diffusion représente la direction et l'importance de la diffusion de l'eau dans le cerveau. La reconstitution mentale d'une image 3D à partir des coupes est difficile. Cela fait que l'utilisation de DTI est presque impossible sans aucune assistance. A cet effet, certaines visualisations du DTI incluent des couleurs pour signaler la direction du flux, allant jusqu'à simuler l'écoulement de l'eau à travers le cerveau et générer des tracts 3D qui représentent les voies neuronales dans le cerveau (voir Figure 2-21).

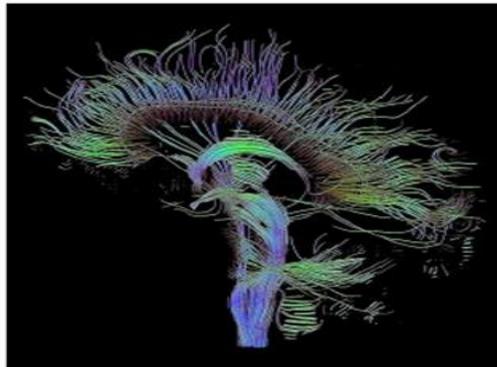


Figure 2-21 : Les voies neurales sont simulées à partir d'images de tenseur de diffusion (Jianu, et al., 2009).

La visualisation de volume devient de plus en plus pratiquée dans de nombreuses applications. Si on pense à la planification d'une intervention chirurgicale, il devient clair combien d'informations utiles peuvent être tirées à partir d'une simple vue 3D des données. Figure 2-22 illustre comment une visualisation 3D correctement conçue peut immédiatement identifier l'emplacement d'un accident vasculaire cérébral ischémique, ou de fournir des détails structurels cruciaux des os fracturés avant une chirurgie reconstructive. Cependant, alors qu'une vue 3D peut être plus intuitive, il a aussi potentiellement l'inconvénient de cacher des informations importantes.

Ainsi, une visualisation 3D efficace doit fournir à la fois une vue dégagée et détaillée de l'objet en question ainsi que de son contexte dans l'espace 3D, tout en gardant une fluidité lors de la navigation. La bonne façon d'atteindre ces objectifs disparates dépend souvent de la situation. Un exemple illustratif est en coloscopie virtuelle, où le

médecin veut trouver des grumeaux sur le côlon appelés polypes. Bien qu'une vue 3D normale peut imiter une vraie coloscopie, les polypes peuvent être trouvés beaucoup plus rapidement en déroulant la paroi du côlon dans une feuille 2D. L'ensemble peut être considéré comme un morceau de papier (voir Figure 2-23).



Figure 2-22 : la visualisation de volume interactif d'une angiographie TDM au diagnostic d'un accident vasculaire cérébral ischémique (gauche) et de planifier la chirurgie reconstructive (droite).

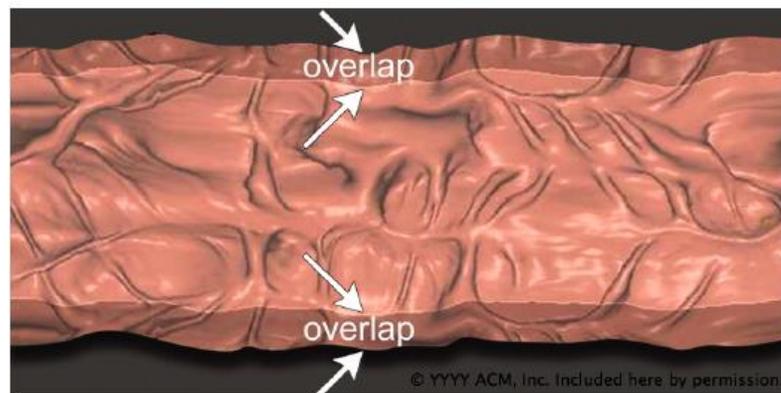


Figure 2-23 : Une vue démantelée du côlon utilisé pour trouver rapidement des polypes dans la coloscopie TDM virtuelle (Hong, et al., 2006).

3. Rendu de surface

3.1. Introduction

Le rendu de surface est l'un des deux principaux modes de rendu des données de volume médical. Au lieu de classer les données de volume et de les afficher directement sur la zone de visualisation, le rendu de surface est basé sur une représentation indirecte du maillage surfacique. Un maillage d'un milieu continu est sa discrétisation (découpage en un nombre fini de parties) spatiale. Autrement dit, c'est une modélisation géométrique de ce domaine (continu) par des éléments proportionnés finis et bien définis. L'objet d'un maillage est de procéder à une simplification d'un système par un modèle représentant ce système et, éventuellement, son environnement (le milieu), dans l'optique de simulations de calculs ou de représentations graphiques (Young, et al., 2008).

En imagerie médicale, ce maillage est généré soit en choisissant une iso-surface à partir des données de volume d'origine ou en utilisant le résultat obtenu après la segmentation.

Le rendu de surface des données d'un volume est principalement motivé par deux points : il est beaucoup plus rapide que le rendu de volume, car un maillage a une occupation mémoire nettement inférieure à celui d'un ensemble de données de volume ; en plus il fournit clairement des images plus réalistes grâce aux indices de profondeur, tels que l'éclairage (voir Figure 3-1).

Le rendu de surface est disponible dans la plupart des stations de travail radiologique et dans la plupart des systèmes de planification de traitement. Les utilisateurs sont familiarisés avec ce mode d'affichage.



Figure 3-1 : Comparaison entre le rendu surfacique et volumétrique au niveau d'éclairage (Kniss, et al., 2002).

Note : Actuellement certaines techniques permettent d'implémenter des algorithmes d'éclairage dans le rendu volumétrique, Ce qui entraîne des calculs plus complexes en raison de la nature globale de l'illumination (Šoltészová, et al., 2011) (Lindemann & Ropinski, 2011).

Cependant, au cours des dernières années, trois développements ont soulevé l'importance de l'extraction de surface et le rendu de surface :

- Avec l'utilisation accrue de la simulation biophysique, il est nécessaire d'avoir un maillage surfacique comme base pour ces simulations.
- Les visualisations 3D Interactive sont utilisées dans les navigateurs. Bien que le rendu de volume direct soit également possible dans de tels contextes, des rendus de surface sont en vigueur en raison de l'exigence de mémoire inférieure.
- La construction de surface devient de plus en plus importante pour l'impression 3D. Pour des questions de planification des traitements compliqués, les parties pertinentes de l'anatomie humaine sont modélisées et imprimées en 3D pour permettre une discussion collaborative approfondie des options de traitement.

3.2. Problèmes topologiques

La topologie de surface est caractérisée par le nombre des composants connectés et leur genre. Le genre d'une surface est le nombre maximum de courbes fermées simples disjointes pouvant être tracées à l'intérieur de cette surface sans la déconnecter (J.Wood, et al., 2004). Un tore, par exemple, est de genre 1, alors qu'une sphère est de genre 0. La plupart des surfaces anatomiques, par exemple, le cerveau, ont un genre de 0. Ainsi, le calcul du genre dans une surface extraite peut indiquer un bruit topologie excessif qui doit être éliminée.

La segmentation se traduit souvent par des contours dans les coupes individuelles. La détermination des surfaces à partir de ces contours est difficile, puisque que trois problèmes doivent être résolus (Meyers, et al., 1992):

- Problème de correspondance : Comment faire la correspondance entre les contours dans la coupe n avec un contour de la coupe $n+1$?
- Problème de carrelage : Comment connecter les points du contour C_n dans la coupe n avec les points du contour C_{n+1} dans la coupe $n + 1$?
- Problème de branchement : Comment diviser le contour C_n du coupe n qui correspond aux contours C_{n+1_a} et C_{n+1_b} de la coupe $n+1$?

Figure 3-2 illustre plusieurs solutions possibles du problème de la correspondance dans le cas où le nombre de contours dans des coupes adjacentes est différent (les contours peuvent être fractionnés ou fusionnés). Bien que le problème de correspondance soit lié au problème de branchement, il peut également se produire dans l'isolement, en particulier si les courbes de contour changent fortement entre les coupes adjacentes.

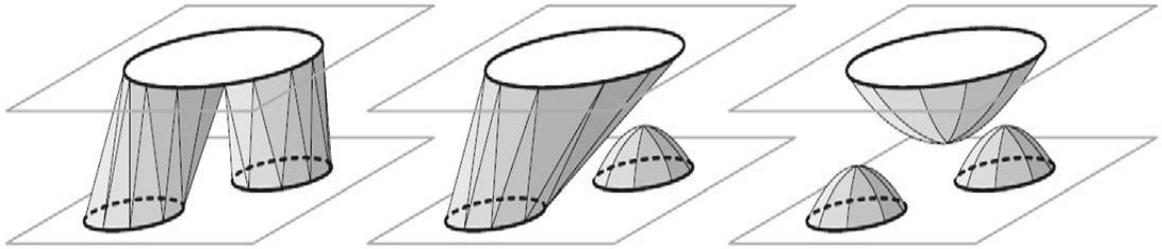


Figure 3-2 : Trois solutions possibles au problème de la correspondance en cas de changement de topologie.

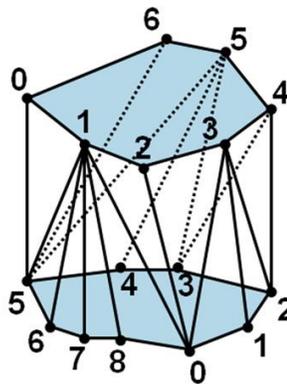


Figure 3-3 : Problème de triangulation de contours de deux coupes.

Note : Le contour dans la coupe n a sept vertex, alors que le contour dans la coupe $n + 1$ a neuf vertex. Il existe dans ce cas de nombreuses triangulations possibles. La bonne triangulation conduit à des objets compacts avec une faible surface et un volume élevé.

Le problème de carrelage est illustré dans la Figure 3-3. Plusieurs auteurs ont discuté à propos de triangulations optimales entre les contours dans les coupes adjacentes (n et $n+1$). Ces problèmes de carrelage sont traités notamment grâce à :

- La maximisation du volume,
- La réduction de la surface,
- La préférence pour une connexion qui optimise la distance entre le barycentre de la coupe n et le barycentre de la coupe $n+1$.

Des solutions efficaces pour résoudre le problème de carrelage sont obtenues à l'aide des processus de recherche dans les graphes (Keppel, 1975) (Fuchs, et al., 1977). Pour cela, une fonction de coût et une combinaison pondérée des trois critères mentionnés ci-dessus sont utilisées pour chercher une solution à travers tous les vertex des deux contours avec un coût minimum.

La topologie d'une surface peut être bien représentée par le graphique de Reeb (Doraiswamy & Natarajan, 2013). Pour construire ce graphique, un balayage aligné sur un axe à travers le volume est effectué pour déterminer les contours de chaque coupe. Chaque contour dans une coupe représente un nœud dans le graphe de Reeb et les arêtes représentent le lien entre les contours dans les coupes adjacentes. Les nœuds ayant plus de deux voisins traduisent un branchement (un contour dans la coupe n correspond à deux contours dans la coupe adjacente). Les cycles dans le graphique Reeb indiquent une structure avec des trous. Les algorithmes de nettoyage de la topologie des surfaces utilisent souvent ce graphique. Dans un maillage triangulaire valide, on ne doit pas avoir des intersections ou des trous dans une surface régulière.

3.3. Représentation de maillages surfaciques

Il existe plusieurs structures de données qui représentent les maillages de surface. Nous décrivons brièvement une structure de données compacte qui est souvent utilisée pour le rendu des maillages de surface. En outre, nous décrivons aussi d'autres structures de données qui fournissent des informations relatives à la topologie d'un maillage triangulaire et donc un accès plus direct à l'information nécessaire pour simplifier, lisser, ou traiter ce maillage triangulaire.

3.3.1. Représentation des vertex partagés

Si le maillage de surface est destiné uniquement pour être rendu (et non traités), une structure de données maigre avec une faible consommation de mémoire est préférée.

Dans ce cas l'utilisation de la représentation des vertex partagés sera optimale. Cette représentation se compose de deux listes :

- une liste de vertex qui comprend les coordonnées de tous ces vertex.
- une liste de faces indexée.

La liste des faces contient des pointeurs vers la liste des vertex, par exemple, avec une face de (5, 6, 8) précise qu'un polygone a des vertex avec les indices 5, 6, et 8 dans la liste des vertex.

Cette représentation indirecte des vertex des polygones dans le stockage a plusieurs avantages par rapport à la représentation directe. Dans un maillage triangulaire régulier, chaque vertex appartient au moins à six triangles. Ainsi, les coordonnées (3 valeurs à virgule flottante à 4 octets) doivent être stockées de façon redondante six fois, au lieu d'un entier représentant l'index. En outre, le système de stockage indirect représente explicitement qu'un certain vertex appartient à six triangles. Si, par exemple, ce vertex est modifié, un seul vecteur 3D représentant les coordonnées a besoin d'être mis à jour.

Si le maillage est constitué uniquement de triangles, chaque triplet d'indices représente un triangle. Dans la forme la plus générale, un ensemble des faces indexé peut contenir de différents types des polygones, par exemple, des triangles et des quadrilatères. Dans ce cas, une valeur spéciale, souvent -1, est utilisé pour identifier les différentes descriptions de polygone.

Pour de nombreux traitements effectués sur des maillages polygonaux, une liste des faces indexée ne fournit pas des informations directes. Par exemple, il est souvent nécessaire de déterminer les deux faces qui partagent une arête dans un maillage. Dans une liste des faces indexées, cela nécessite une recherche dans l'ensemble de la structure des données. Il serait plus commode, si ces relations de voisinage sont directement stockées. Idéalement, toutes les relations adjacence entre les faces, les arêtes et les vertex sont explicitement représentées.

Plusieurs sous-ensembles de ces relations sont représentés dans les structures de données, telles que la structure Arête-Ailée (Baumgart., 1975) et de la structure à demi-arête (Botsch, et al., 2002) que nous expliquons dans la suite.

3.3.2. Structure de données Arête-Ailée

La structure de données Arête-Ailée est centrée sur les arêtes. Dans cette structure de données, chaque arête a un pointeur vers deux sommets et sur les deux faces adjacentes. De plus, une arête, comme partie d'une chaîne des arêtes, a un pointeur à son successeur et son prédécesseur (voir Figure 3-4). Cette structure de données ne sont pas idéales pour les algorithmes qui effectuer des itérations sur les sommets du maillage et de les modifier en fonction de leur voisinage, puisque les voisins de sommets ne peuvent pas être directement accessibles.

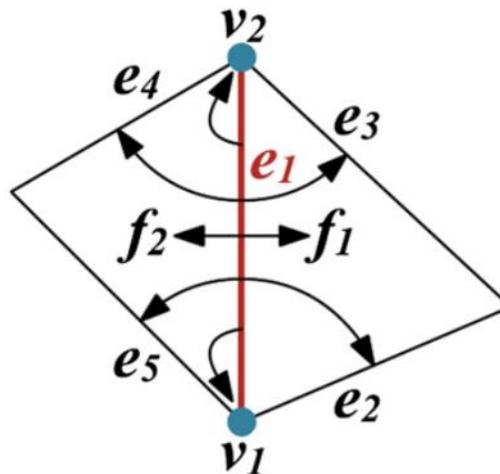


Figure 3-4 : structure de données Arête-Ailée.

Note : Un arête $e1$ comporte des pointeurs vers ses deux sommets ($v1$, $v2$), aux arêtes qui partagent ces sommets ($e2$, $e3$, $e4$, $e5$), et les deux faces qui contiennent $e1$.

3.3.3. Structure des données Demi-arêtes

La structure à demi-arête divise chaque arête en deux arêtes avec les mêmes coordonnées mais avec une direction opposée. Chaque demi-arête pointe vers sa demi-arête opposée, le sommet incident, le prochain et précédent demi-arête de la même face, et le polygone incident (voir Figure 3-5).

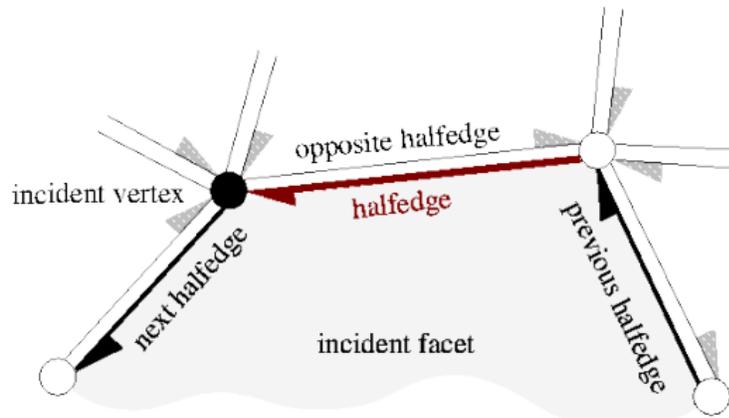


Figure 3-5 : Demi-arête et relations d'incidence stockées.

Les structures à base Demi-arêtes ont été prises par CGAL¹ et OPENMESH² pour représenter un maillage. Pour les algorithmes qui nécessitent l'insertion ou la suppression d'éléments, listes chaînées doubles sont prévus, alors que les représentations matricielles sont favorables à des algorithmes qui ne modifient pas les éléments de maillage existants. (Botsch, et al., 2010) fournissent un bon aperçu sur le traitement des maillages et les structures de données connexes.

¹ <http://www.cgal.org>

² <http://www.openmesh.org>

3.4. DICOM et le stockage du maillage

3.4.1. Standard DICOM

Les données d'image médicale sont physiquement stockées ensemble avec des informations indispensables pour l'interprétation de ces images. Ces informations sont standardisées à la suite des activités de normalisation de longue date partant de 1985. Ces activités ont conduit notamment à la norme DICOM (Digital Imaging and Communications in Medicine) qui a été créé par la NEMA (National Electrical Manufacturers Association).

La version actuelle 3.0 a été créée en 1993. Cette version ouverte aux actualisations, a connu depuis lors bon nombre d'améliorations et de nouveautés.

Parmi les sous versions notables, signalons en particulier, celle qui a eu lieu en 2008 avec la prise en compte de la représentation sous forme de maillage en ce qui concerne les images 3D

En parallèle, les nouveaux outils logiciels d'imagerie médicale se sont imposés de prendre en charge dans leurs présentation cette norme afin d'être acceptés et soutenus par un plus large public.

DICOM est la norme de l'industrie d'imagerie médicale pour le transfert des images radiologiques et d'autres informations médicales entre les ordinateurs et les appareils médicaux. Au cours des dernières années, DICOM a été étendu pour saisir également les exigences des autres disciplines médicales, y compris de nouveaux types de données d'image, par exemple, les données de microscopie acquises dans les départements de pathologie.

DICOM permet la communication numérique entre les équipements de diagnostic et de thérapeutique et des systèmes de différents fabricants. Par exemple, les postes de travail, scanners TDM et IRM, numériseurs de films, archives partagées et les ordinateurs centraux provenant de différents fournisseurs peuvent communiquer au moyen de DICOM.

Actuellement, la plupart des fabricants des scanners TDM et IRM sont officiellement crédités comme étant « DICOM conformant », ce qui signifie que leur production est conforme aux spécifications de la norme DICOM.

3.4.2. Structure des données DICOM

DICOM contient des informations concernant divers aspects des données d'images médicales. Ces informations sont organisées dans une hiérarchie dont le patient est le sommet. Pour chaque patient une ou plusieurs études sont disponibles; chaque étude peut contenir plusieurs séries et enfin chaque série peut contenir des instances. Ces instances peuvent représenter une ou plusieurs images, des informations sur des marqueurs fiduciaires, des données de recalage par rapport à un référentiel, des documents associés, etc. Ces informations sont en général optionnelles. Il en est de même d'un fichier nommé « DICOMDIR » qui peut se trouver dans le dossier patient et qui fournit une description globale d'une ou plusieurs séries (voir Figure 3-6).

L'instance qui nous intéresse dans cette partie de la thèse est celle qui représente le maillage d'une surface d'un volume des données médicales (Treichel, et al., 2011) détaillée dans la suite de ce même chapitre.

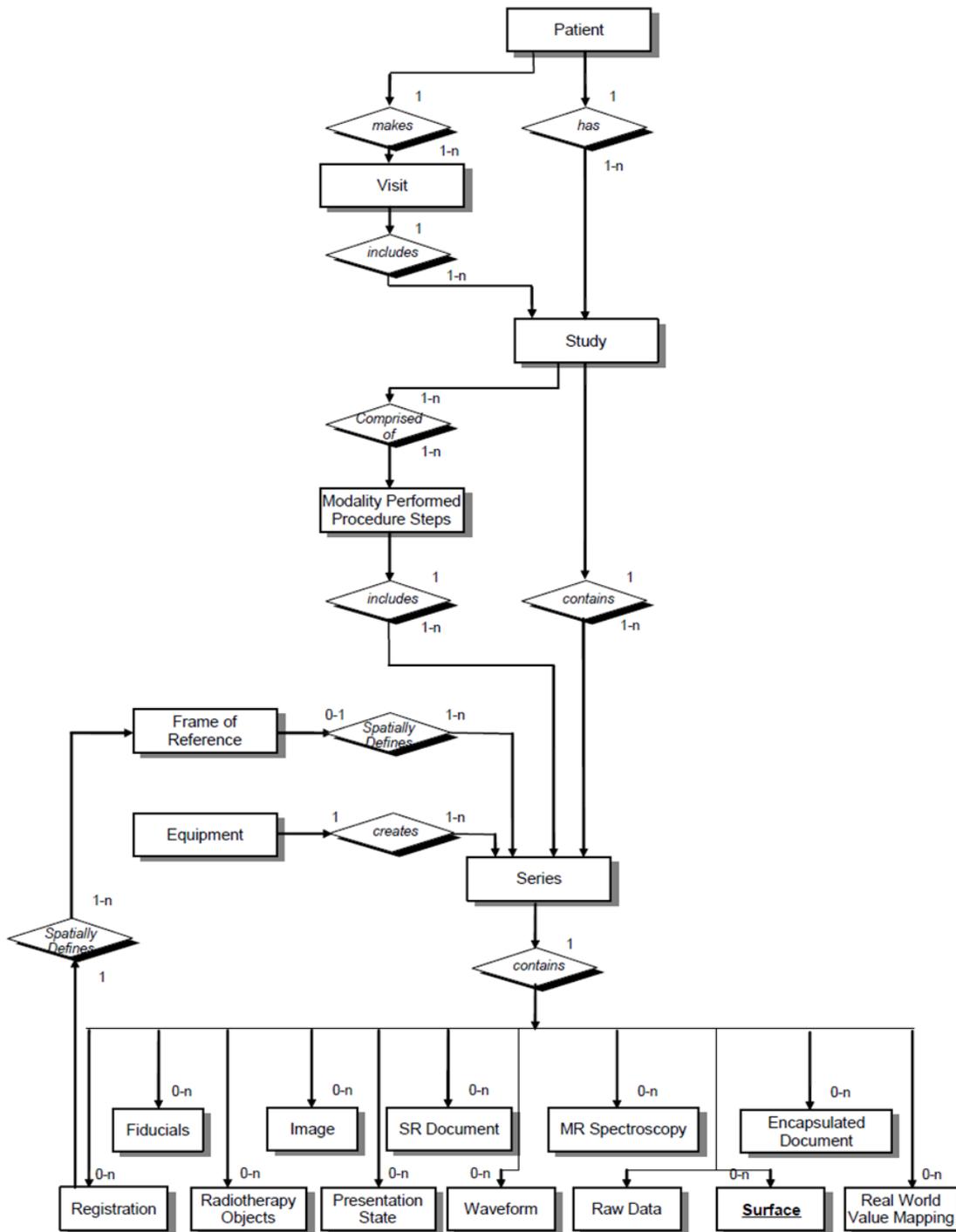


Figure 3-6 : Représentation du modèle DICOM et des étapes médicales équivalentes.

3.4.3. Supplément DICOM de description polygonal d'une surface

DICOM a été en permanence affiné par son comité qui comprend 26 principaux fabricants ainsi que plusieurs organisations, telles que l'American College of Radiology, qui représentent les utilisateurs. En plus de ce comité, 27 groupes de travail traitent de sujets spéciaux tels que l'information cardiaque et vasculaire, la dentisterie, la médecine nucléaire, la pathologie, la chirurgie et la mammographie numérique. Le groupe de travail le plus récemment fondé traite la technologie Web. Ces groupes se composent de représentant de l'industrie, de la pratique clinique et du milieu universitaire. Les principaux résultats d'un groupe de travail sont représentés sous forme des suppléments qui permettent de déterminer comment résoudre un problème spécifique.

Le groupe de travail DICOM 24 « DICOM chirurgicale », donne des précisions sur l'utilisation efficace des maillages surfaciques 3D, ce qui a conduit au supplément 132 ("Surface Segmentation Storage SOP Class") qui a été accepté en 2008. Ce supplément est particulièrement important pour la visualisation médicale 3D et à la chirurgie assistée par ordinateur (Treichel, et al., 2011).

Le Supplément introduit un nouveau module de maillage surfacique qui est utilisé pour encoder les données d'une segmentation de surface. La classe de ce supplément peut spécifier une surface dérivée de toute modalité agréée DICOM ou d'une technique de mesure non-DICOM ou peut également être utilisé pour spécifier les surfaces conçues.

Le supplément prend en charge les fonctionnalités suivantes:

- Les données de segmentation sont encodées dans une représentation polygonale de la surface.
- Une ou plusieurs segmentations est prise en charge par instance.
- La position et l'orientation sont définies dans le système de coordonnées spécifié par le cadre de référence (Frame of Reference).

- Une instance de recalage spatial est utilisée pour localiser la position des surfaces sur d'autres instances.

Exemples De Codage

Tableau 3-1 illustre l'utilisation de supplément 132 pour présenter le maillage d'un tétraèdre (voir Figure 3-7).

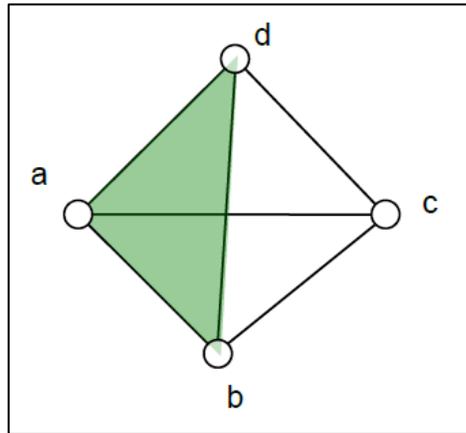


Figure 3-7 : Maillage de Tétraèdre.

Tag	Name	Value	Comment
(0066,0001)	Number of Surfaces	1	
(0066,0002)	Surface Sequence		
(0066,0003)	Surface Number	1	
(0066,0004)	Surface Comments	Test Surface	
(0066,0009)	Surface Processing	YES	
(0066,000a)	Surface Processing Ratio	1.0	
(0066,000b)	Surface Processing Description	Moved Object	
(0066,0035)	Surface Processing Algorithm Identification Sequence		
0066,002F)	Algorithm Family Code Sequence		
(0008,0100)	Code Value	1231009	
(0008,0102)	Coding Scheme Designator	DCM	
(0008,0104)	Code Meaning	Manual Processing	
(0066,0030)	Algorithm Name Code Sequence		
(0008,0100)	Code Value	AA01	
(0008,0102)	Coding Scheme Designator	ICCAS	
(0008,0104)	Code Meaning	Interactive Shift	
(0066,0036)	Algorithm Name	Interactive Shift	
(0066,0031)	Algorithm Version	"V1.0"	

	(0066,0032)	Algorithm Parameters	“x = 5 y = 1 z = 0”	
	(0062,000C)	Recommended Display Grayscale Value	FFFFH	
	(0062,000D)	Recommended Display CIE Lab Value	FFFF\8080\8080	
	(0066,000c)	Recommended Presentation Opacity	1.0	
	(0066,000d)	Recommended Presentation Type	SURFACE	
	(0066,000e)	Finite Volume	YES	
	(0066,0010)	Manifold	YES	
	(0066,0011)	Surface Points Sequence		
	(0066,0015)	Number Of Points	4	
	(0066,0016)	Point Coordinates Data	-5.\ -3.7\ -4.7\5.\ -3.7\ -4.7\ 0.\ 7.4\ -4.7\0.\ 0.\ 8.3	
	(0066,0017)	Point Position Accuracy	0.001\0.001\0.001	
	(0066,0018)	Mean Point Distance	10.0	
	(0066,0019)	Maximum Point Distance	10.0	
	(0066,001A)	Points Bounding Box Coordinates	-5.\ -3.7\ -4.7\ 5.\ 7.4\ 8.3	2 triplets
	(0066,001B)	Axis of Rotation	0.0\0.0\1.0	
	(0066,001C)	Center of Rotation	0.0\0.0\0.0	
	(0066,0012)	Surface Points Normals Sequence	<empty>	
	(0066,0013)	Surface Mesh Primitives Sequence		
	(0066,0025)	Vertex Point Index List	<empty>	
	(0066,0024)	Edge Point Index List	<empty>	
	0066,0023)	Triangle Point Index List	1\3\2\1\2\4\2\3\4\3\ 1\4	
	(0066,0026)	Triangle Strip Sequence	<empty>	
	(0066,0027)	Triangle Fan Sequence	<empty>	
	(0066,0028)	Line Sequence	<empty>	
	(0066,0034)	Facet Sequence	<empty>	

Tableau 3-1 : Présentation d’un tétraèdre selon le supplément 132

3.5. Marching cubes

Dans cette partie de ce chapitre nous expliquons le principe de l’algorithme de Marching cubes, support d’une grande partie de notre contribution. La présentation que nous avons choisie d’en faire respecte l’évolution chronologique de cet algorithme ou plutôt de cette classe d’algorithmes ; d’un point de vue pédagogique le lecteur comprendra mieux les subtilités à cerner afin de mieux comprendre notre travail.

Lorsqu’on s’intéresse à l’extraction de l’isosurfaces à partir des données d’image volumétriques, parmi les nombreux algorithmes disponibles (Johnson & Hansen, 2004),

il y a un algorithme classique: le Marching Cubes (Lorensen & Cline, 1987). Essentiellement, l'algorithme de Marching Cubes s'intéresse à des cellules de volume. Ces dernières sont obtenues après l'application d'un cadrillage aux coupes voisines et elles sont délimitées par des éléments de surface du cadrillage. Le Marching Cubes examine individuellement chaque cellule de volume et génère une triangulation au cas où l'isosurface d'intérêt intersecte la cellule de volume. Ainsi, le procédé d'extraction de l'isosurface d'un volume est décomposé en cellules individuelles qui sont traitées localement (voir Figure 3-8).

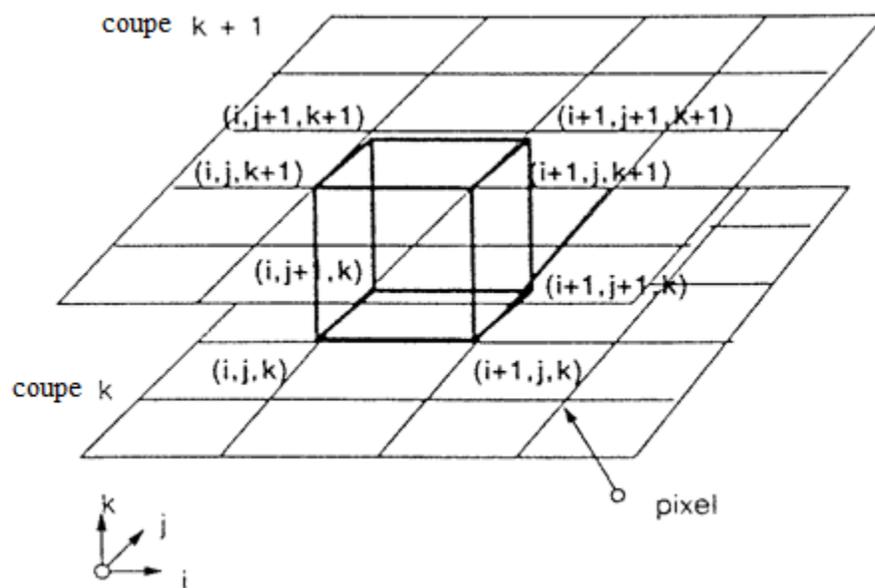


Figure 3-8 : Extraction de l'isosurface à l'aide de Marching cubes en traitant localement les cellules de chaque coupe.

Une cellule est constituée de huit points d'échantillonnage. Dans un ensemble de données avec $N \times M \times O$ éléments, il existe $N-1 \times M-1 \times O-1$ cellules à traiter. L'innovation majeure de Marching Cubes est l'utilisation d'une table de consultation (la table de cas) pour chaque triangulation possible. Cela permet d'avoir une triangulation nettement plus rapide de l'isosurface spécifiée. Cependant, puisque le Marching Cubes fonctionne localement, les incohérences de la triangulation entraînent parfois des problèmes dans le maillage final.

Pour présenter ce principe, nous allons d'abord discuter de la version 2D du Marching Cubes (Marching Squares). Ensuite, nous reviendrons au Marching Cubes 3D lui-même.

3.5.2. Marching squares

L'algorithme de Marching Squares calcule l'iso-lignes à partir des images 2D (Schroeder, et al., 2001). Essentiellement, Marching Squares examine toutes les cellules d'image. Pour chaque cellule (carré), nous examinons les valeurs des sommets : si elles sont supérieures ou égales à l'iso-valeur (ou seuil) le sommet est considéré interne à la zone d'intérêt. Ce qui génère au total 16 interprétations possibles au niveau des carrés (interne, externe, intersecte avec 1 des sommets, 2 sommets, ...).

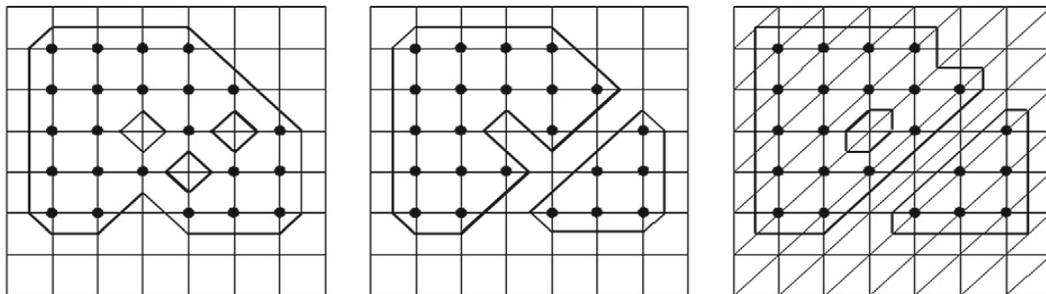


Figure 3-9 : les contours possibles d'un ensemble de pixels.

Note : De gauche à droite : Marching Square contourne avec la solution connectée ; contourne avec la solution séparée ; contourne après la triangulation (Marching Triangles). Tous les résultats sont valides, mais ils sont topologiquement différents et le nombre de trous et de composants est différent.

Si nous codons les états -d'appartenance ou non- des sommets de chaque cellule carrée comme des chiffres (0 ou 1) d'un nombre, ce nombre (comme par exemple 0100) peut servir comme index à une table qui énumère les 16 cas pour les lignes de contour. La

plupart de ces cas sont très similaires. Avec les rotations et l'inversion des états de coins, nous pouvons réduire le nombre de cas au cinq configurations possibles.

Dans cette stratégie, deux cas restent ambigus, car on ne peut pas en déterminer exactement les contours sans l'apport d'information supplémentaire :

Dans ces cas, deux états de coins opposés ont la valeur 1 et les deux autres ont la valeur 0 (les deux coins sont situés en première diagonale ou non). Ces deux ensembles peuvent soit être connectés ou séparés. La Figure 3-9 (à gauche, au centre) montre l'impact du cas choisi sur le contour résultant. Un remède pour éviter l'ambiguïté est d'utiliser au départ un type de cellule différente, par exemple, une triangulation, ce qui génère un contour unique, (voir Figure 3-9, à droite), alors que la transformation des cellules quadrilatérales en triangles ne sont pas uniques.

3.5.3. Algorithme de base 3D

Nous allons maintenant examiner l'algorithme Marching Cubes (Lorenson & Cline 1987). Cet algorithme se répartit en cinq étapes pour extraire une surface à partir d'un ensemble des données de volume:

- Détermination de l'indice de cas de chaque cellule : A partir de l'étude d'appartenance des sommets de la cellule à l'isosurface
- Détermination des arêtes d'intersection : les arêtes d'intersection sont les arêtes du cube reliant sur une surface élémentaire les deux coupes.
- Calcul des intersections au moyen d'une interpolation linéaire.
- Triangulation des intersections.
- Calcul du vecteur normal des surfaces pour l'éclairage.

Le Marching Cubes traite chaque cellule de volume indépendamment.

Nous développerons, dans la suite, les démarches adoptées pour réaliser ces différentes étapes

3.5.3.1. Détermination de l'indice de cas et des arêtes d'intersection

L'indice de huit voxels de la cellule (sommets du cube) de volume est calculé, cela indique si leur valeur respective est supérieure, égal ou inférieure à l'iso-valeur spécifiée τ .

L'état de chaque voxel est interprété comme un chiffre binaire (la valeur 1 indique que le voxel se situe à l'intérieur de volume des données. Dans le cas inverse, on aura la valeur 0), ce qui donne un indice sous forme d'un nombre de huit bits (voir Figure 3-10, à gauche)).

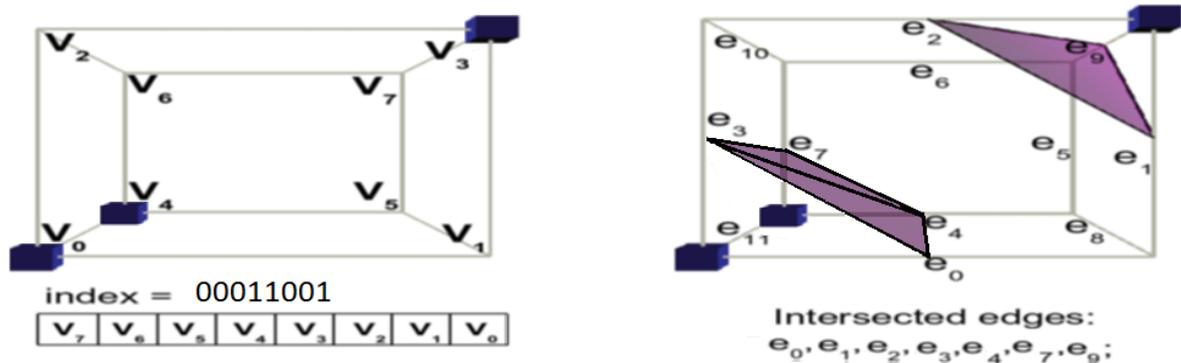


Figure 3-10 : Indexation des Voxels et des arêtes de l'algorithme Marching Cubes.

Note : A gauche : Pour calculer l'indice de la table des cas, les valeurs des voxels composent un indice de huit bits, suivant si elles sont fixées à 0 ou 1. Dans ce cas, l'indice de ce cube est 00011001 en base 2, ce qui précise que seuls les sommets V0, V3 et V4 sont internes à la zone d'intérêt. A droite : La table de cas contient une liste des arêtes coupées. Dans cet exemple, les arêtes e0, e1, e2, e3, e4, e7, e9 sont coupées. Les cubes au niveau des positions de voxels indiquent que leurs valeurs sont fixées à 1.

Une fois l'index de cas de la cellule est déterminé, les arêtes de cellules qui sont traversées par l'isosurface peuvent être recherchées dans la table de cas. Dans la Figure 3-10 à droite, sept arêtes de cellules sont coupées par l'isosurface.

Remarquons que nous avons maintenant 256 configurations possibles (au lieu des 16 en 2D) des états de voxels, et par conséquent, nous avons 256 triangulations possibles d'une cellule de volume. Cependant, toutes ces configurations permettent d'avoir différentes triangulations. Les cas peuvent être classés en 15 classes d'équivalence, en tenant compte de la rotation ou la mise en miroir.

La Figure 3-11 représente les 15 classes comme indiqué par (Lorenson & Cline, 1987). Cette réduction était très utile aux débuts, lorsque le Marching Cubes a été inventé et moins de nos jours: en fait, il permet de stocker l'état de deux cellules dans les huit bits d'un octet, ce qui réduit efficacement la consommation de stockage de la table de cas. La Figure 3-11 montre, également, qu'au maximum quatre triangles sont générés comme contribution locale d'une cellule.

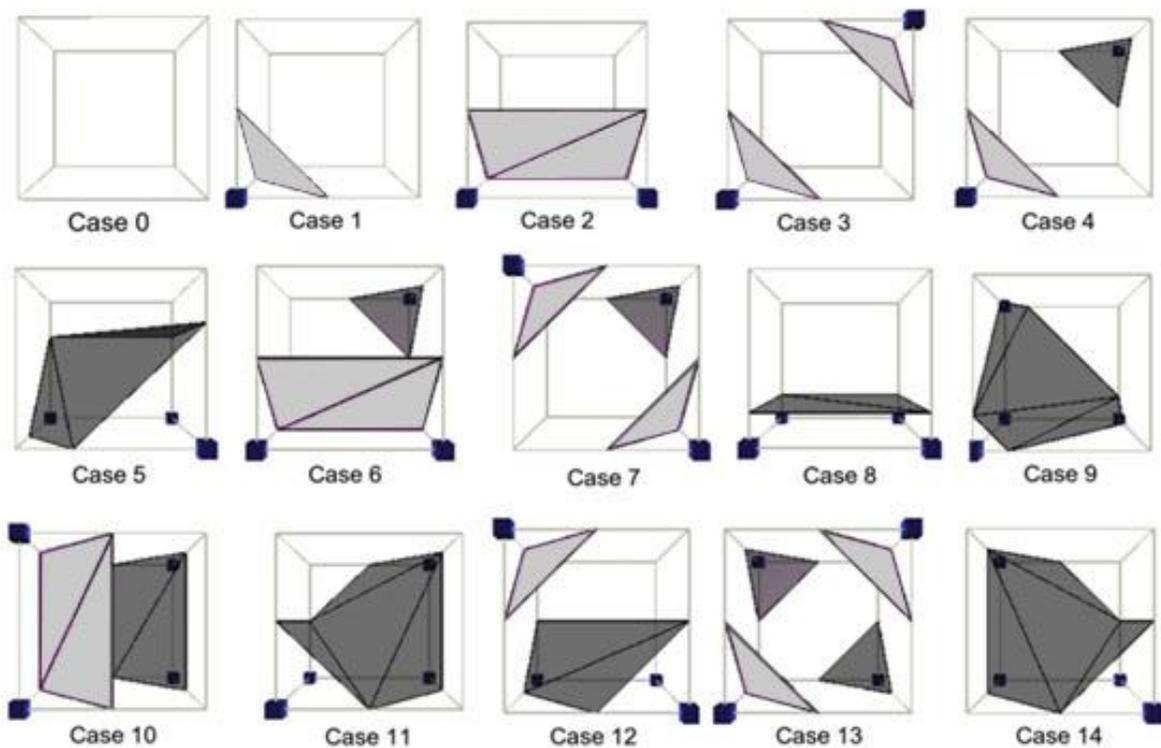


Figure 3-11 : 15 classe des cas de Marching Cubes.

Note : Les triangles gris clair sont les faces avant et les triangles gris foncés sont les faces arrière. Les cubes au niveau des positions de voxels indiquent la valeur 1 (tous les autres voxels ont la valeur 0).

3.5.3.2. Calcul des intersections

Une fois que les arêtes intersectées de la cellule sont déterminées, l'intersection elle-même est calculée par une interpolation linéaire. Le paramètre d'interpolation t est calculé en se basant sur l'isovaleur τ , les voxel V_j et V_{j+1} sur les deux côtés de l'arête selon l'équation 3.1. Si on suppose que X_j et X_{j+1} décrivent les coordonnées respectives des voxels, on peut calculer l'intersection X_e avec une interpolation linéaire selon l'équation 3.2.

$$t = \frac{\tau - V_j}{V_{j+1} - V_j} \quad (3.1.)$$

$$X_e = X_j + t \cdot (X_{j+1} - X_j) \quad (3.2.)$$

Si l'iso-valeur τ est plus proche de la valeur de voxel V_j , le point d'intersection de l'arête se rapproche de ce voxel. Respectivement, le point d'intersection de l'arête s'éloigne de ce voxel, si τ est plus proche de la valeur de voxel V_{j+1} . Si t est exactement égal 0 ou 1, la surface passerait par un voxel de la cellule. Cela provoque des problèmes dans la triangulation. C'est pourquoi t est souvent légèrement modifié pour donner une triangulation valide.

3.5.3.3. Triangulation des Intersections

Dans la table des cas, nous avons des informations sur la façon de la composition des intersections des arêtes en triangles, donc après l'interpolation des intersections, on peut déterminer les triangles.

Enfin, les normales de surface doivent être calculées pour permettre une visualisation ombragée qui illustre clairement la forme correspondante. Les normales de surface sont déterminées en fonction de l'orientation du triangle et sont directement exploitées par les GPU des cartes graphiques actuelles.

Cependant, dans ce contexte nous pouvons approcher les gradients du champ scalaire en calculant les différences d'intensité. L'utilisation des données initiales du champ scalaire conduit d'ailleurs à une définition plus précise des normales de surface.

Ce calcul est effectué pour les trois sommets d'un triangle et la solution simple est de calculer leur moyenne. Notons enfin que le gradient obtenu doit être normalisé pour l'utiliser dans un modèle d'illumination.

3.5.4. Limites et améliorations de Marching cubes

Dans ce qui précède, nous avons fourni le principe du Marching cube tel qu'il a été initié par ses concepteurs. Dans ce qui suit, nous discutons des limites remarquées et signalées de Marching Cubes ainsi que les améliorations qui lui ont été apportées.

Comme tout procédé d'extraction de surface, Marching Cubes est basé sur la décision binaire ; si des portions d'une cellule appartiennent à la surface ou non. Ainsi, la valeur d'isosurfaces est déterminée selon la surface à extraire, telles que les dents, la peau et les os (voir Figure 3-12).

Toutefois, le résultat final peut contenir des erreurs s'il est généré à partir des données bruitées et inhomogènes ou lorsque les phénomènes intervenants sont amorphes avec des limites plus floues.

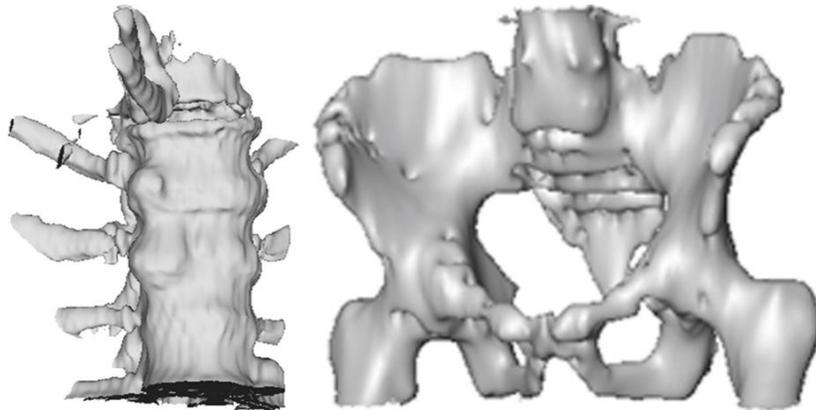


Figure 3-12 : Iso-surfaces extraites à partir des données TDM avec Marching Cubes

3.5.4.1. Problèmes de topologie

D'abord, nous discutons le Marching Cubes en relation avec les trois problèmes expliqués dans la détermination des surfaces à partir des contours (voir § 3.2).

Marching Cubes ne résout pas le problème de correspondance. Les surfaces des coupes adjacentes sont connectées uniquement lorsqu'il y a un chevauchement de ces surfaces dans ces coupes. En règle générale, cette situation est la plus fréquente, en particulier pour les gros objets compacts et une « faible » distance entre les coupes. Toutefois, si l'une de ces deux conditions n'est pas remplie, le problème de la déconnection des composants de surface peut apparaître dans le résultat final (voir Figure 3-13).



Figure 3-13 : Un résultat de Marching Cubes à partir d'une segmentation d'un vaisseau sanguin allongé, partiellement fortement courbé.

Note : Les contours dans les coupes adjacentes ne sont pas toujours en chevauchement. Par conséquent, Marching Cubes ne peut pas générer une surface connectée; L'ensemble des données a une résolution relativement faible entre les coupes (3 mm), par rapport à une résolution dans le plan de 0,5 mm.

La solution du problème de carrelage est acceptable mais pas optimale. Toute solution optimale au problème de la correspondance et du carrelage nécessite de traiter l'ensemble de contour de deux coupes adjacentes au lieu de le traiter localement dans des

cellules individuelles. Ainsi, Marching Cubes offre un compromis entre la qualité et la vitesse. Il s'exécute beaucoup plus rapidement que les autres méthodes qui privilégient l'optimisation globale.

Le problème de branchement n'est pas considéré explicitement, car pratiquement, il n'y a pas de solutions générales pour aborder ce problème d'une façon idéale autrement qu'en modifiant les paramètres d'acquisition des données.

3.5.4.2. Ambiguïtés en 3D

Similaire au cas 2D, Marching Cubes souffre d'ambiguïtés. Selon la façon dont les intersections sont triangulées : connecter ou séparer les surfaces. Les ambiguïtés peuvent également provoquer des incohérences, comme des trous (voir Figure 3-14). Ici, on a échec de l'hypothèse que l'extraction de l'isosurface peut être effectuée localement et que la solution obtenue est erronée et non seulement ambiguë (comme dans le cas 2D). Les iso-surfaces corrects sont connectés tant qu'ils ne traversent pas la frontière de l'ensemble de données. Néanmoins, Ce problème, ainsi que d'autres, relatif à l'algorithme de Marching Cubes a été détecté dès 1988 (Dürst., 1988) et un certain nombre de solutions ont été proposées.

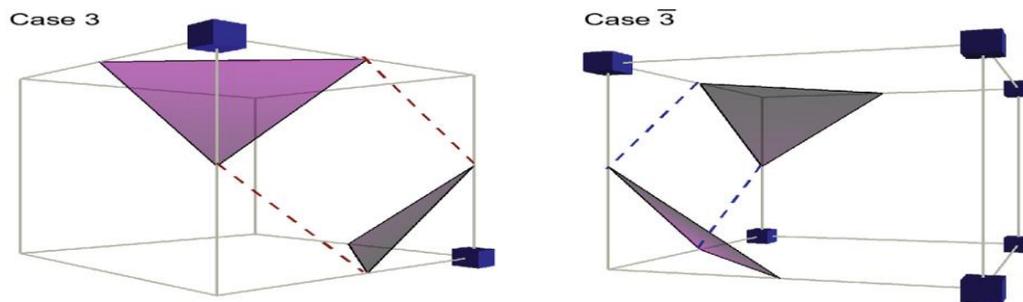


Figure 3-14 : Les triangulations possibles de Marching Cubes pour deux cellules de volumes voisins.

Note: Nous représentons deux cellules voisines. La cellule de gauche est triangulée, les voxels ayant la valeur 1 d'une manière séparée (cas $\bar{3}$), la cellule voisine de droite est triangulé les voxels ayant la valeur 1 d'une manière connectée (cas 3). Les deux triangulations sont valides, mais elles génèrent une interface triangulaire incompatible avec un trou. Respectivement, Les lignes pointillées indiquent les arêtes du triangle de l'autre cellule.

Le premier problème détecté dans l'algorithme de Marching Cubes est les cas d'ambiguïtés trouvés dans certaine configuration, pour le même cube on peut avoir plusieurs façons de trianguler. Nielson et Hamann ont identifié le problème d'ambiguïté de face qui se produit lorsqu'on a deux sommets diagonalement opposés marqués positif et les deux autres marqués négatifs (Nielson & Hamann, 1991). Cette ambiguïté peut conduire à des trous au niveau de topologie. Natarajan et Chernyaev, indépendamment, ont montré un autre type d'ambiguïté qui se produit à l'intérieur d'un cube (Natarajan, 1994) (Chernyaev, 1995).

Ces ambiguïtés peuvent souvent être résolues en ajoutant des points de test supplémentaire dans chaque cube. Plusieurs méthodes sont proposées pour résoudre ce problème, soit pour les cas ambigus sur les faces (Nielson & Hamann, 1991), soit à l'intérieur des cubes (Lopes & Brodlie, 2003) (Nielson, 2003).

L'algorithme de Marching Cubes divise l'espace en une grille cubique régulière, la résolution de la surface polygonale obtenue dépend directement de la taille de cette grille. L'augmentation du nombre de cubes dans la grille permet d'augmenter la résolution de la surface polygonale, mais le nombre de triangles résultant pourrait être important même si la surface originale est assez simple. Afin de réduire le nombre de triangles, plusieurs méthodes ont été mises au point pour appliquer l'algorithme de Marching Cubes sur une grille adaptative. (Shu, et al., 1995) proposent une méthode qui introduit le concept de la génération multi-résolution de la grille, algorithme qui adapte la taille des triangles avec la forme de la surface. (Congote, et al., 2009) proposent une autre méthode qui préserve la même taille de la grille, il augmente la résolution de la surface en déplaçant les sommets de la grille vers la surface du cube, ce qui augmente le nombre de cubes contenant la surface.

L'algorithme de Marching Cubes ne permet pas d'extraire les arêtes vives « Sharp features » même en diminuant la taille de la grille. (Kobbelt, et al., 2001) propose la marche cube étendue (EMC), ils solutionnent ce problème en détectant la présence des arêtes vives par l'analyse de l'orientation des normales à la surface au niveau des intersections avec les arêtes de chaque cube traité, un autre travail présenté par (Ju, et al.,

2002) qui se base sur les contours doubles (DC), c'est une méthode hybride d'EMC et surfaceNets (Gibson, 1998), il permet de conserver les arêtes vives et d'éviter les trous et les fissures dans une résolution adaptative.

Ce qu'on peut reprocher à l'ensemble de ces contributions qui ont cherché à dépasser les problèmes de Marching Cubes est que la plupart d'elles se concentrent uniquement sur un seul problème isolé. Par la suite, la combinaison entre toutes ces méthodes n'est pas simple ni évidente. Parmi les algorithmes qui proposent de résoudre tous ces problèmes à la fois, on a Cubical Marching squares (CMS), cet algorithme est relativement simple et facile à mettre en œuvre, l'idée essentiel de cette méthode se base sur la conversion de la Marching Cubes 3D en six faces 2D, chaque face est traitée indépendamment, les arêtes générées sur ces faces sont remises en 3D pour former les triangles de la surface, cette méthode permet de préserver les arêtes vives, de maintenir une topologie cohérente et de générer une surface adaptative (Ho, et al., 2005).

4. Unité de traitement graphique

Comme nous l'avons annoncé dans l'introduction générale, les contributions de cette thèse résident dans l'exploitation de nouvelles (partant de 2008-2012) architectures des processeurs graphiques des ordinateurs domestiques. Ce chapitre sert à initier le lecteur au domaine relativement nouveau qui concerne les nouvelles fonctionnalités de GPU. Les algorithmes développés dans cette thèse se basent sur la **tessellation** matérielle de GPU. Une discussion sur cette nouvelle architecture de GPU s'impose donc avant de plonger dans les détails de notre contribution.

4.1. Introduction

A partir de 2003, date à laquelle la vitesse du GPU est devenue plus grande que celle du CPU, passant par les années 2008 à 2012 au cours desquelles les principaux développeurs de GPU comme NVIDIA, ATI et INTEL ont commencé à intégrer les composants permettant de raffiner les maillages « tessellation ». Les développeurs de bibliothèque graphiques comme DirectX et OPENGL ont rendu disponibles bon nombre de nouvelles fonctionnalités permettant de mettre à profit les nouvelles technologies. L'explosion qui a eu lieu dans l'industrie des jeux vidéo a d'ailleurs alimenté d'une manière croissante et inédite l'avancement matériel de ces processeurs graphiques et celui des bibliothèques de logiciels correspondant.

Comme indiqué par NVIDIA, La carte graphique TITAN XP intègre 3840 processeurs de flux et fonctionne à une performance de plus de 12 Téra Flops (*floating-point operation per second*) avec une bande passante mémoire de 547.7 Go/s. Bien que les CPU commencent à profiter d'un traitement multi-core, ainsi, cette vitesse est encore à un ordre de 700 Giga Flops avec une bande passante mémoire de 34,1 Go/s sur un processeur Intel® Core™ i7-7920HQ. Alors, pourquoi la performance des processeurs GPU a augmenté plus rapidement que celle des processeurs CPU au cours de ces dernières années ? La différence est principalement due à des différences majeures dans l'architecture de ces deux plates-formes.

Les processeurs CPU sont conçus pour avoir des performances élevées sur les algorithmes séquentiels, avec une grande proportion de transistors dédiés à l'exécution hors-ordre et à la prédiction de branchement. La nature très parallèle des problèmes graphiques a permis aux GPU d'utiliser un pourcentage beaucoup plus élevé de transistors sur le débit arithmétique.

En outre, les GPU peuvent exploiter l'augmentation du nombre de transistors très efficacement en ajoutant simplement des noyaux de traitement. Cela a abouti à un taux de croissance qui est nettement plus rapide que la loi de Moore

Theoretical GFLOP/s

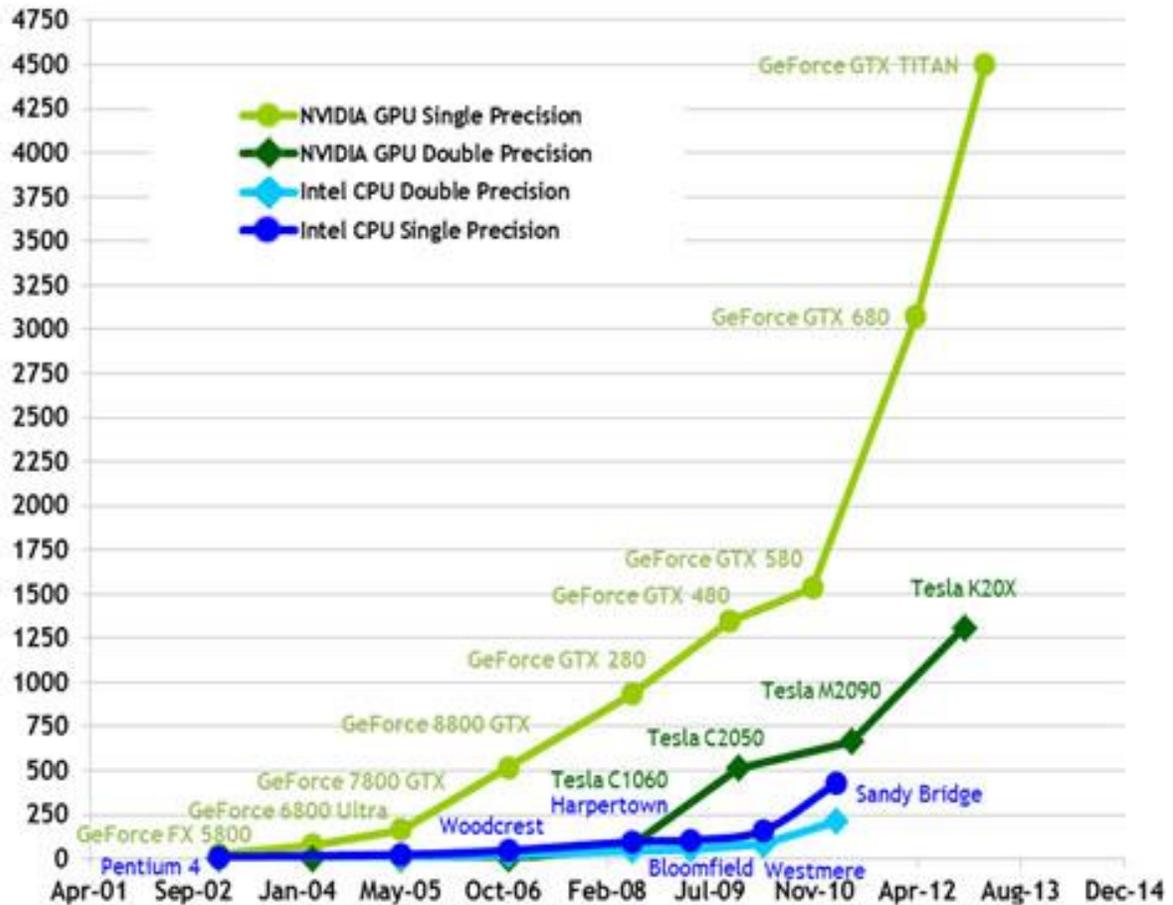


Figure 4-1 : l'évolution de la puissance de calcul des CPUs et GPUs.

Les mesures sont en Flops, ce qui correspond au nombre d'opérations à virgule flottante par seconde. Il est donc facile de constater une évolution très forte de la puissance de calcul des GPUs en comparaison à celle des CPUs. Que ce soit en simple ou double précision, les GPUs ont une puissance de calcul très importante.

Puisque presque tous les ordinateurs domestiques mis en vente aujourd'hui sont dotés d'un processeur graphique, l'ubiquité du GPU et son faible coût en font une alternative très convaincante aux superordinateurs et aux postes de travail coûteux.

4.2. Shaders

4.2.1. Programmation des cartes graphiques

Les interfaces de programmation graphique sont utilisées pour exploiter directement le matériel graphique connecté à un ordinateur. Les deux APIs graphiques les plus populaires sont OpenGL et DirectX. Dans le passé, ces APIs graphiques proposaient un ensemble d'algorithmes et des états de rendu qu'un programmeur pouvait activer ou désactiver en tout moment dans une application. Cet ensemble d'algorithmes et d'états de rendu est connu sous le nom de pipeline à fonction fixe. Bien que le pipeline à fonction fixe offre une certaine commodité lorsqu'il s'agit d'exécuter des algorithmes spécifiques supportés par l'API, son inconvénient majeur est qu'il est restrictif dans le nombre de fonctionnalités et il lui manque des qualités de « personnalisation » permettant aux développeurs d'écrire leurs propres algorithmes et de les exécuter dans le matériel graphique. Les shaders programmables sont la solution aux limitations et restrictions du pipeline à fonction fixe d'une API graphique.

Un shader est un code qui peut être exécuté sur le matériel graphique. Un shader programmable est un moyen pour les développeurs d'écrire des algorithmes personnalisés qui peuvent fonctionner sur les données qui composent leurs scènes virtuelles.

Ces programmes sont appelés shaders car ils sont souvent utilisés pour contrôler les effets d'éclairage et d'ombrage (voir Figure 4-2).

Actuellement, Les shaders peuvent être utilisés pour créer à peu près n'importe quel effet imaginable, ce qui donne aux développeurs un haut niveau de liberté et de flexibilité indépendamment de l'API utilisée.

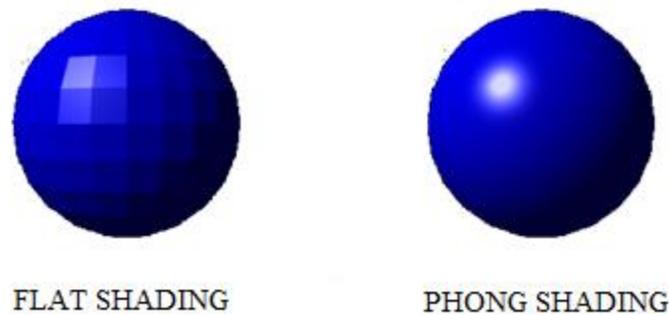


Figure 4-2 : Les shaders sont souvent utilisés pour produire l'éclairage et l'ombre dans la modélisation 3D.

Note : L'image de gauche fournit un rendu de sphère en utilisant un ombrage de l'API standard, alors que celle de droite est obtenue grâce à un shader programmé

Aujourd'hui, il existe quatre différents types de shaders qui peuvent être utilisés pour réagir avec les différentes informations qui composent une scène virtuelle. Ces shaders sont vertex shaders, Geometry shaders, Tessellation shaders, Pixel shaders (voir § 4.3.2). Lorsqu'il est combiné en un effet, un ensemble de shaders est collectivement appelé un programme de shader.

4.2.2. Shaders de bas niveau

Aux débuts des shaders programmables, les développeurs devaient utiliser un langage d'assemblage pour coder les effets qu'ils voulaient utiliser dans les applications graphiques.

La plupart des programmeurs qui connaissent le langage d'assemblage admettent qu'il est plus difficile de développer et de maintenir un code basé sur l'assemblage qu'un langage de haut niveau comme C ou C ++, bien que vous puissiez avoir beaucoup de contrôle sur les instructions sur un niveau inférieur. En outre, les deux APIs graphiques principales, OpenGL et Direct3D, avaient des langages machines légèrement différents, de sorte qu'un shader écrit pour l'un ne fonctionne pas avec l'autre. Cela signifie que les

développeurs devaient créer des effets non seulement pour les différentes versions de shader qu'ils souhaitaient prendre en charge, mais aussi pour les deux APIs.

4.2.3. Shaders de haut niveau

Les langages de shaders de haut niveau offrent une alternative flexible aux langages de shaders de bas niveau. Un langage de shaders de haut niveau basé sur un langage populaire, comme C, permet aux développeurs qui travaillent déjà avec ce langage d'écrire, de lire et de maintenir plus facilement le code shader.

Les langues de haut niveau possèdent plusieurs avantages par rapport aux langues de bas niveau :

- Plus facile à lire
- Plus facile à modifier
- Plus facile de suivre les erreurs
- Plus facile à écrire
- Plus rapide à développer, car les shaders de bas niveau prennent plus de temps en général que ceux de haut niveau

Le principal avantage des langages de bas niveau est que les développeurs ont le contrôle ultime sur la façon dont les registres de la machine sont utilisés, alors que dans un langage de haut niveau, un compilateur a le rôle de générer le code de bas niveau. Certains compilateurs sont très efficaces, mais cela ne conduit pas toujours au code le plus rapide et le plus efficace possible.

La nécessité de développer un langage de haut niveau est commencée dès la sortie du pipeline graphique programmable. NVIDIA a pris l'initiative pour créer Cg, tandis que Microsoft, qui a participé à la création de Cg, a développé HLSL, et OpenGL avait GLSL. Chacun de ces langages de shaders de haut niveau est basé sur C, et ils sont très semblables.

4.3. Pipeline graphique de rendu

Avant de pouvoir étudier et implémenter notre technique de visualisation 3D d'un volume des données médicales, il est nécessaire d'étudier l'architecture du pipeline graphique adopté conjointement par les développeurs des bibliothèques graphiques et les fabricants des puces graphiques.

4.3.1. Processeur graphique

Une scène typique qui doit être rendue sous forme de graphiques 3D est composée de nombreux objets distincts. Les formes géométriques de ces objets sont représentées par un ensemble de sommets et un type particulier de primitive graphique qui indique comment les sommets sont connectés pour produire une forme. La Figure 4-3 illustre les dix types de primitifs graphiques définis par la bibliothèque OpenGL. Le matériel graphique est capable de rendre un ensemble de points individuels, une série de segments ou un groupe de polygones. La plupart du temps, la surface d'un modèle 3D est représentée par une liste de triangles, dont chacun renvoie trois points dans une liste de sommets.

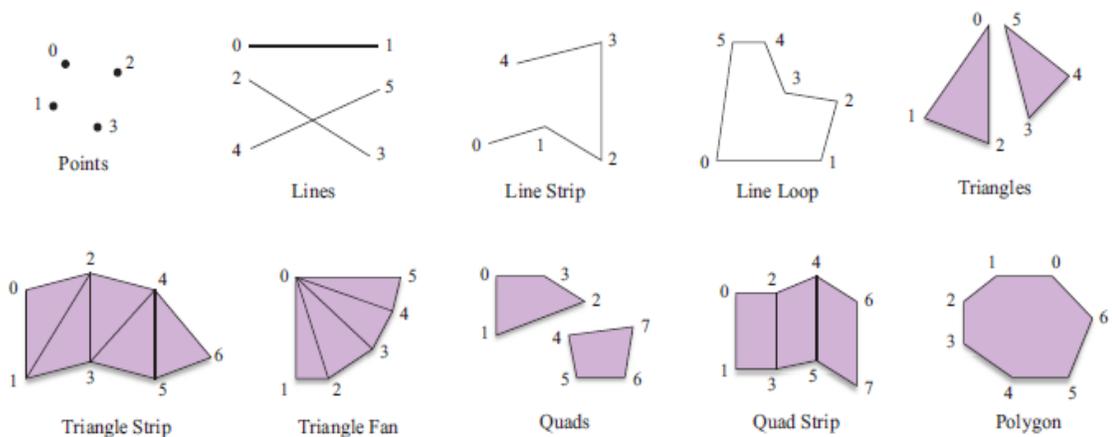


Figure 4-3 : Les dix types de primitives définies dans OpenGL.

La carte graphique actuelle possède une unité de traitement graphique (GPU) dédiée qui exécute des instructions indépendamment de l'unité centrale de traitement

(CPU). Le CPU envoie des commandes au GPU afin d'exécuter les instructions de rendu alors que le CPU continue avec d'autres tâches. C'est ce qu'on appelle un fonctionnement de sous-traitance asynchrone.

Une application pilote le GPU à l'aide des commandes d'une bibliothèque de rendu, telle que OpenGL. A son tour cette bibliothèque envoie des commandes à un driver qui sait comment communiquer au GPU dans son langage natif (instructions machine). L'interface graphique (ici OpenGL) s'appelle couche d'abstraction de matériel ou HAL (Hardware Abstraction Layer). Cette interface nous donne un ensemble commun de fonctions qui peuvent être utilisées pour rendre une scène sur n'importe quel matériel graphique compatible avec l'architecture OpenGL. Le driver traduit les appels des fonctions OpenGL en code compréhensible par le GPU. La Figure 4-4 présente un diagramme qui montre les communications qui se déroulent entre le CPU et le GPU.

Une carte graphique possède sa propre mémoire, qui est généralement appelée VRAM (Video Random Access Memory). Le GPU peut stocker des informations sur la VRAM, mais il existe certains types de données standards qui peuvent presque toujours être trouvées dans la mémoire de la carte graphique lorsqu'une application graphique 3D est en cours d'exécution. Parmi ces données, il y a les tampons d'image (Image Buffers) avant et arrière. Le tampon d'image avant contient les données de pixels exactes qui sont visibles sur l'écran et le tampon d'image arrière reçoit les pixels qui sont en train d'être calculés pour la prochaine image. Une fois qu'une image a été complètement rendue, les tampons d'image avant et arrière sont échangés.

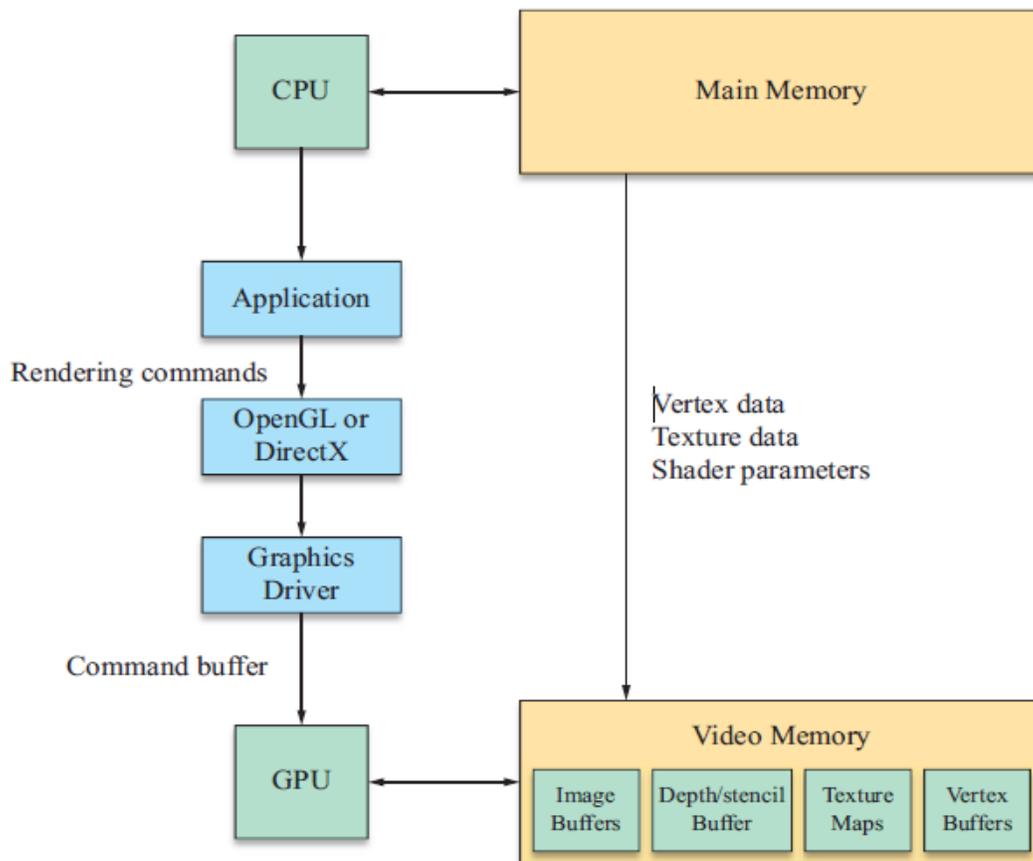


Figure 4-4 : Communication entre le CPU, le GPU et les mémoires associées.

VRAM stocke aussi un bloc de données appelé le tampon de profondeur ou z-buffer. Le tampon de profondeur stocke, pour chaque pixel dans le tampon d'image, une valeur qui représente la mesure de profondeur du pixel dans l'image. Le tampon de profondeur est utilisé pour effectuer l'élimination de la surface cachée en autorisant uniquement la création d'un pixel si sa profondeur est inférieure à la profondeur du pixel déjà dans le tampon d'image.

Il existe un autre tampon appelé le tampon stencil qui sera créé avec les tampons d'image et le tampon de profondeur. Ce tampon sert parfois à masquer des portions de l'image ou à d'autres traitements particuliers, comme la gestion des ombres par exemple.

Pour la grande majorité des applications de rendu 3D, l'utilisation de VRAM est dominée par des cartes de texture. Les cartes de texture sont des images qui sont appliquées à la surface d'un objet pour lui donner plus de détails dans la visualisation.

Dans les applications de rendu avancées, les cartes de texture peuvent contenir des informations autres qu'une simple image de pixel. Par exemple, cette carte peut contenir des vecteurs qui représentent des pentes variables à différents endroits sur la surface d'un objet (bump map).

4.3.2. Les étapes de pipeline graphique

Dans une carte graphique, le flux des données qui décrit une scène virtuelle est un ensemble de polygones planaires. Le GPU est conçu pour produire des images rasters, qui correspondent à l'ensemble de pixels formant l'image, à partir de cet ensemble de polygones. Ce procédé est généralement constitué d'une succession d'opérations assurées par un pipeline dans lequel des données progressent les unes derrière les autres pour augmenter le débit en décomposant les calculs en plusieurs parties. Ce pipeline graphique est doté des unités programmables qui permettent aux programmeurs d'implémenter leurs programmes afin de créer des effets personnalisés.

La nomenclature utilisée pour la description de différentes étapes de pipeline graphique est basée sur la bibliothèque d'OpenGL.

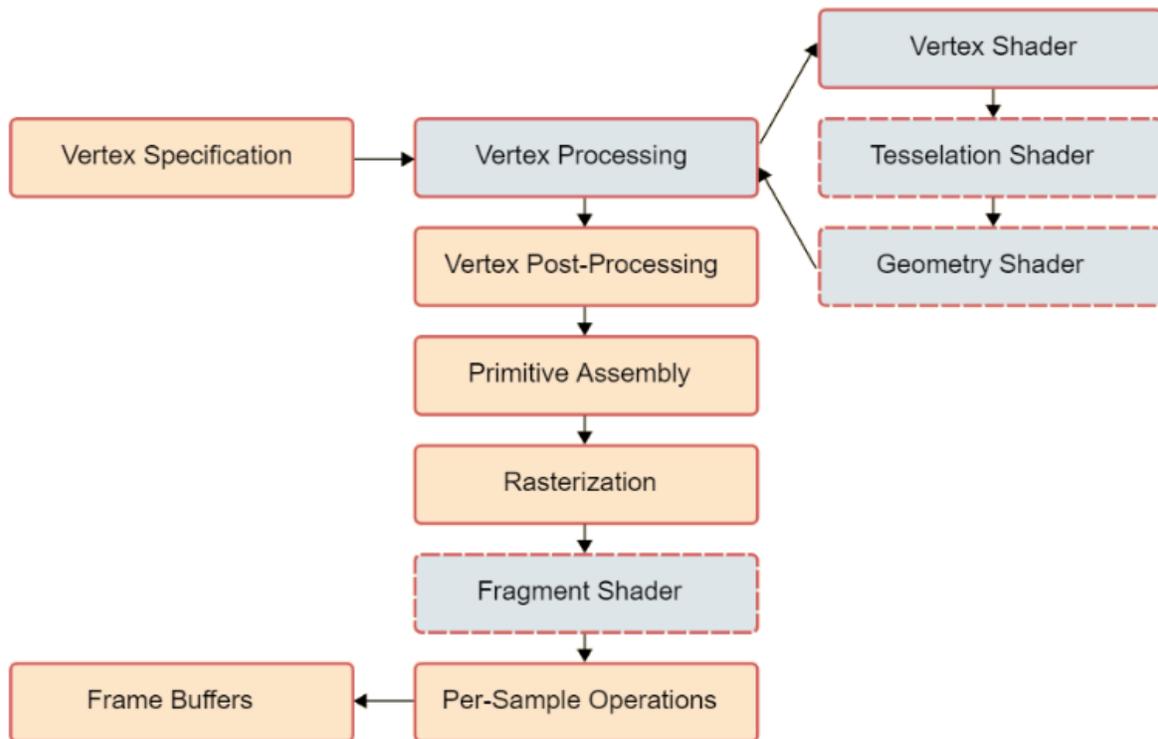


Figure 4-5 : Diagramme du pipeline graphique de rendu programmable

Le pipeline de rendu fonctionne dans l'ordre suivant :

- Spécification des sommets

Le processus de la spécification des sommets est déterminé par la mise en place d'une liste ordonnée de sommets à envoyer au pipeline. Ces sommets définissent les limites d'une primitive. Les primitives sont des dessins basiques, comme des triangles, des lignes et des points.

- Traitement des sommets

Les étapes de traitement des sommets sont des étapes programmables. Cela permet aux utilisateurs de personnaliser la façon dont ces sommets sont traités. À cette fin trois unités sont disponibles :

Vertex shader	<p>Les shaders Vertex effectuent un traitement de base individuel sur chaque sommet. Les shaders Vertex reçoivent les attributs (coordonnées du sommet, couleur, vecteur normal, ou autres) des sommets de l'étape précédente et associe à ce sommet entrant un seul sommet sortant en se basant sur un programme défini par l'utilisateur. Les shaders Vertex peuvent avoir des sorties définies par l'utilisateur, mais il existe également une sortie spéciale qui représente la position finale du sommet.</p>
Tessellation	<p>Les primitives peuvent être tessellées en utilisant deux étapes et un tessellateur à fonction fixe entre elles. L'étape <i>Tessellation Control Shader</i> (TCS) détermine en premier la quantité de tessellation à appliquer à une primitive, tout en assurant la connectivité entre les primitives tessellées adjacentes. L'étape <i>Tessellation Evaluation Shader</i> (TES) est en dernier, elle applique l'interpolation ou d'autres opérations utilisées pour calculer les valeurs de données définies par l'utilisateur pour les primitives générées par le processus de tessellation à fonction fixe. Vu son importance dans notre thèse, cette étape sera encore plus explicitée à la fin de ce même chapitre.</p>
Geometry Shader	<p>Les géométries shaders sont des programmes définis par l'utilisateur. Ces programmes sont capables de supprimer les primitives, ou les tesseller en donnant de nombreuses primitives pour une seule entrée. Les géométries shaders peuvent convertir des primitives en différents types; Les primitives des points d'entrée peuvent devenir des triangles, ou les lignes peuvent devenir des points.</p>

- Post-traitement des sommets

Après le traitement du sommet basé sur le shader, les sommets subissent à un certain nombre d'étapes de traitement à fonction fixe. :

Feedback de transformation	Les sorties de géométrie shader sont écrites dans une série d'objets tampons qui ont été configurés à cet effet. Elles permettent à l'utilisateur de manipuler les données obtenues via les vertex et géométrie shaders, en les stockant et en permettant de les utiliser ultérieurement.
Decoupage	Le découpage signifie que les primitives situées sur la limite entre l'intérieur et l'extérieur du volume visionné sont divisées afin de garder seulement les parties de primitives qui seront affichées sur l'écran. Les positions de vertex sont transformées en coordonnées de la fenêtre.

- Assemblage primitif

L'assemblage primitif est le processus consistant à collecter les attributs des sommets de sortie des étapes précédentes et à les assembler afin de construire une séquence de formes primitives. Le type de primitive choisi par l'utilisateur détermine la manière dont ce processus fonctionne.

La sortie de ce processus est une séquence de primitives simples (lignes, points ou triangles...). Si par exemple, l'entrée est une primitive en triangle strip (voir Figure 4-3) contenant 12 sommets, par exemple, la sortie de ce processus sera de 10 triangles.

Face culling (élimination de primitives)

Des primitives peuvent être éliminées et ainsi rejetées sans rendu en fonction de leur orientation dans les coordonnées des fenêtres. Cela permet d'éviter de rendre les primitives qui ne sont pas dans le champ visuel de l'utilisateur observateur. Pour les

surfaces fermées, de telles primitives seraient naturellement recouvertes par des primitives tournées vers l'utilisateur; il n'est donc jamais nécessaire de les rendre.

- Rasterisation

Les primitives qui atteignent cette étape sont alors rasterisées dans l'ordre dans lequel elles ont été données. Le résultat de la rasterisation d'une primitive est une séquence de Fragments. Un fragment est un ensemble d'états (comme la couleur et la profondeur) d'un pixel, utilisé pour calculer les données finales dans le buffer framebuffer.

- Fragment shader

Les données de chaque fragment de l'étape précédente sont traitées par un fragment shader. La sortie du fragment shader est une liste de couleurs, une valeur de profondeur et une valeur de stencil. Les fragments shaders ne sont pas en mesure de définir les données de stencil pour un fragment, mais elles ont un contrôle sur les valeurs de couleur et de profondeur.

- Per-Sample Operations

Les données de fragment donné par le processeur de fragments passent ensuite une séquence d'étapes :

- La première étape est une séquence de tests d'élimination, lorsqu'un test est actif et que le fragment échoue, les pixels sous-jacents ne sont pas mis à jour. Ces tests sont : Pixel ownership test, Scissor Test, Stencil Test, Depth Test. Pour plus de détail sur ces test le lecteur peut consulter par exemple (Ginsburg, et al., 2014).
- Après cela, intervient l'étape de mélange des couleurs. Pour chaque valeur de couleur de fragment, il existe une opération de mélange spécifique entre elle et la couleur déjà dans le frame buffer à cet emplacement.

- Enfin, les données du fragment sont écrites sur le frame buffer. Les opérations de masquage permettent à l'utilisateur d'empêcher des écritures sur certaines valeurs. Les écritures de couleur, de profondeur et de stencil peuvent être désactivées ; Des canaux de couleurs individuels peuvent également être masqués.

4.4. Tessellation

La tessellation est une nouvelle unité de shader intégrée dans le pipeline graphique de GPU. Cette unité permet de subdiviser des primitives simples en maillages plus fins au moment de l'exécution selon une fonction mathématique.

4.4.1. Principe de tessellation

Les cartes graphiques d'aujourd'hui sont composées de processeurs parallèles allant jusqu'à plusieurs milliers de cœurs (Guide, 2013). Cependant, alors que les GPU ont une puissance de calculs très importante de données brutes, ils sont principalement limités par la bande passante mémoire. Un goulot d'étranglement se forme en particulier pour les rendus en temps réel notamment lorsqu'une géométrie de surface très détaillée doit être mise à jour (par exemple, pour l'animation) et pixellisée dans chaque image. Pour faire face à ce problème, la tessellation matérielle a été introduite avec la Xbox 360 (ANDREWS & BAKER, 2006) et l'API DirectX11 (BOYD, 2008).

L'idée clé est de générer une géométrie très détaillée à partir d'une représentation basse résolution. Par conséquent, les maillages sont définis comme un ensemble de patches, plutôt que d'une représentation purement triangulaire. Au moment de l'exécution, les patches sont envoyés aux processeurs du GPU, où ils sont directement raffinés et pixellisés sans utiliser les E/S mémoire. La densité de Tessellation est spécifiée selon les besoins, ce qui permet d'avoir des systèmes plus flexibles au niveau de détail. En outre, des détails géométriques à haute fréquence peuvent être ajoutés à la volée en déplaçant les

sommets générés. Cela permet d'avoir des animations à faible coût puisque seuls les points d'entrée de patch qui doivent être mis à jour.

La tessellation matérielle est largement utilisée dans les jeux vidéo pour l'affichage extrêmement détaillés des objets animés. Dans l'industrie de l'animation, la tessellation matérielle a également été identifiée comme une technique utile pour la modélisation interactive. Actuellement, la tessellation matérielle est également disponible sur les appareils mobiles (Alben, 2013) (QUALCOMM, 2013), ouvrant la porte à de nouvelles applications en graphisme mobiles.

4.4.2. Étapes de tessellation matérielle

Le pipeline graphique typique sur les GPU actuels se compose de cinq étapes de shader programmable (voir Figure 4-6).

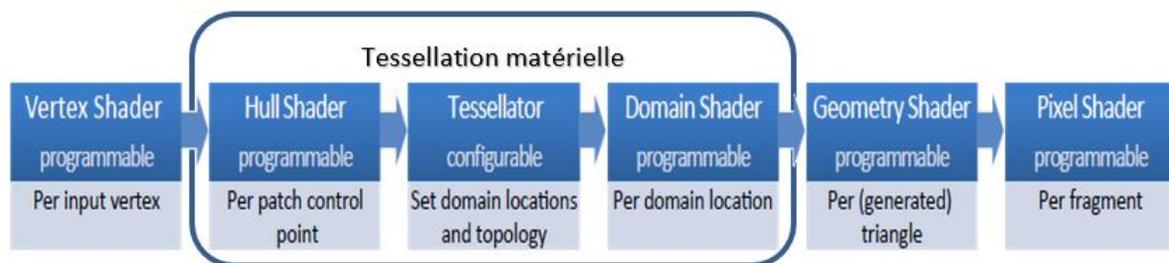


Figure 4-6 : Pipeline graphique selon la nomenclature DirectX 11 comportant les étapes shaders programmables et matériels configurables.

La tessellation matérielle a été accessible aux développeurs dans DirectX depuis DirectX 11 (BOYD, 2008) et OpenGL depuis la version 4.0 (Segal, 2012). Dans ce qui suit, nous allons utiliser la nomenclature DirectX.

Les maillages fortement-tessellés entraînent une grande réservation de la mémoire du GPU et par conséquent le rendu devient plus coûteux. La tessellation matérielle permet la possibilité d'avoir une grande quantité de polygones en sortie car l'accès à la mémoire globale est uniquement nécessaire pour les points de contrôle d'entrée. (Nießner, et al., 2016) ont effectué une référence à cela par la génération de 2 millions de polygones

sur une carte graphique NVIDIA GTX 780 en utilisant des grilles planaires, puis ils ont comparé les résultats obtenus entre le rendu traditionnel et la tessellation matérielle. Le rendu classique qui utilise un tampon de sommet dure environ 1,5 ms, par contre le rendu de la même quantité de polygones à l'aide de la tessellation matérielle ne prend que 0,25 ms, c'est-à-dire plus efficace par un facteur de 6. Cependant, l'activation de la tessellation matérielle sans la tessellation du maillage d'entrée est inefficace, car le rendu de 2 millions de triangles avec l'unité de tessellation et le traitement de chaque triangle comme un maillage séparé, ralenti le rendu 5 fois plus. Par conséquent, la tessellation matérielle ne doit être utilisée qu'en cas de nécessité.

La tessellation matérielle introduit trois nouvelles phases en pipeline graphique entre le vertex shader et la géométrie shader (voir Figure 4-6): l'étape Hull shader, l'étape Tessellator et l'étape domain shader.

4.4.2.1. Étape Hull shader

L'étape Hull shader est à son tours divisée en deux parties logiques: une fonction constante per-patch et le programme hull shader.

La fonction constante per-patch est exécutée une fois par patch d'entrée et cette fonction est utilisée pour déterminer la densité de la tessellation du patch. A ce titre, les facteurs de tessellation per-patch sont calculés et envoyés à la fonction fixe de l'étape Tessellator afin de préciser la quantité des points d'échantillonnage à générer.

Pour les domaines triangulaire (tri) et quadrangulaire (quad), il y a des facteurs de tessellation pour les bordures (dont 3 pour tri et 4 pour quad) et l'intérieur (dont 1 pour tri, 2 pour quad). Le domaine isoligne est seulement contrôlé par deux facteurs de tessellation qui déterminent la densité et le détail de la ligne.

Le programme hull shader est exécuté pour chaque point de contrôle de patch. Les données de tous les points de chaque patch d'entrée sont partagées entre les programmes hull shader du même patch. Actuellement, le nombre de points de contrôle est limité à 32 par patch.

4.4.2.2. Tessellator étape

L'étape hull shader est suivie par la fonction fixe tessellator, cette fonction génère les points d'échantillonnage et la topologie pour un patch donné en fonction des facteurs de tessellation d'entrée.

Des exemples de modèles de tessellation obtenus sont présentés dans la Figure 4-7 et la Figure 4-8, comprenant les domaines triangulaire et quadrangulaire.

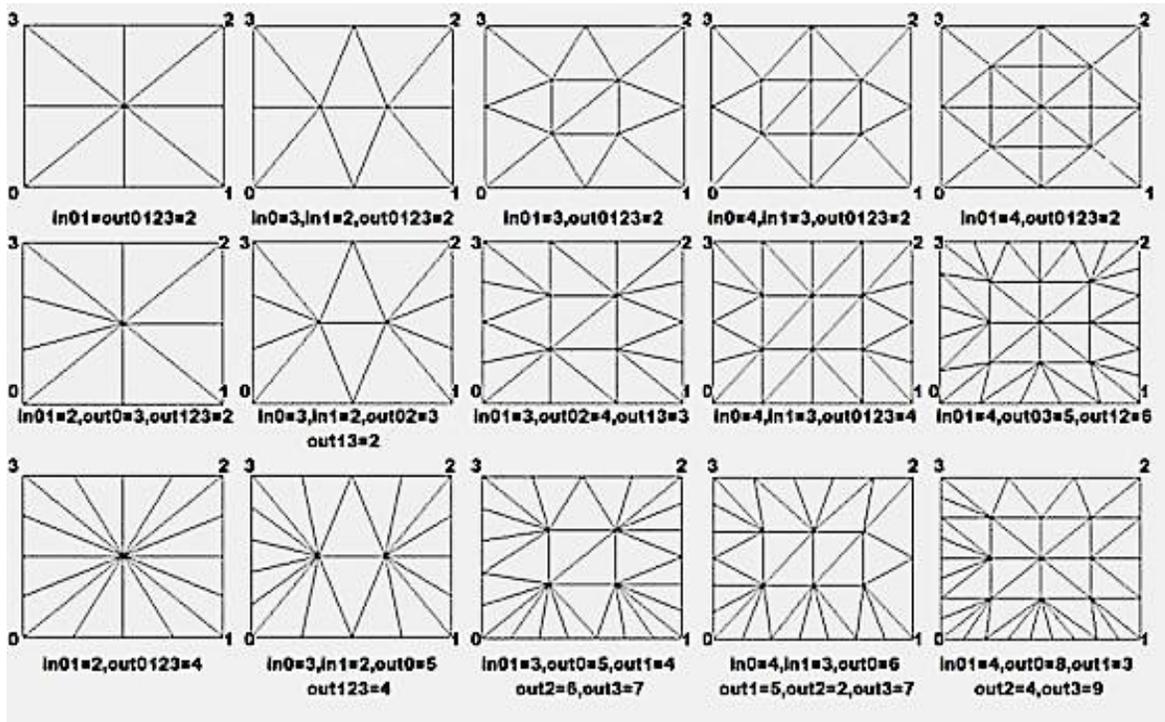


Figure 4-7 : Création des modèles de tessellation d'un quad pour différents facteurs de tessellation.

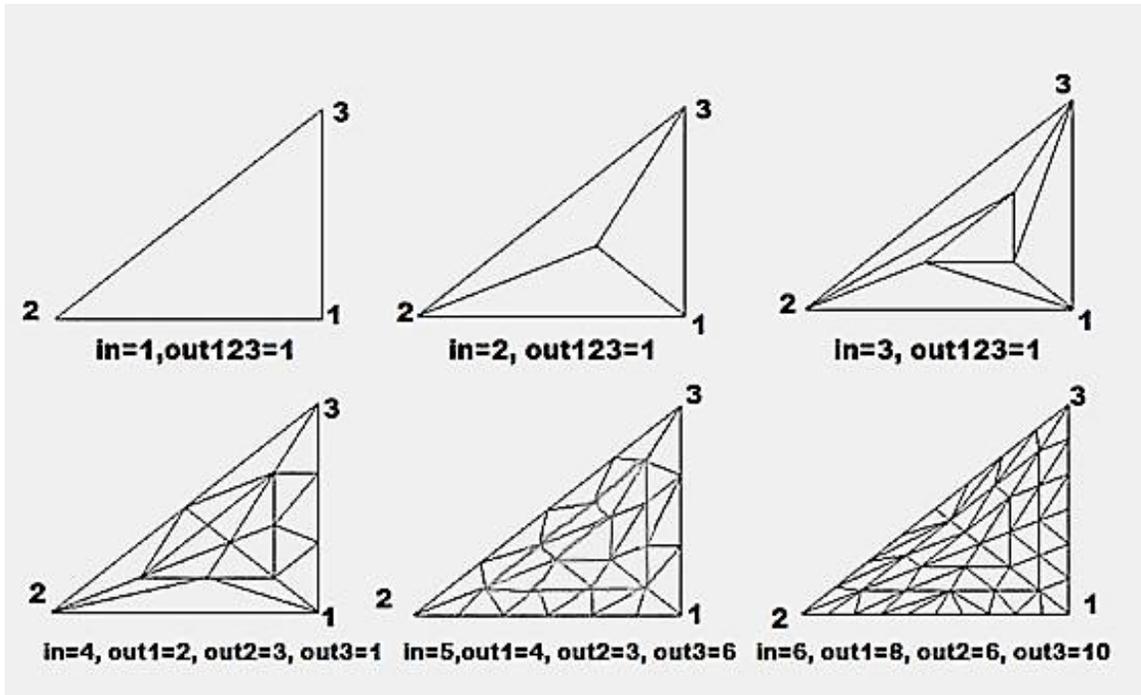


Figure 4-8 : Création des modèles de tessellation d'un triangle pour différents facteurs de tessellation.

4.4.2.3. Domain shader étape

La dernière étape du pipeline de la tessellation est l'étape programmable domain shader où un programme shader est invoqué pour chaque point d'échantillonnage.

L'entrée de cette étape est constituée de :

- La sortie de l'étape hull shader : les facteurs de tessellation, les points de contrôle.
- La sortie de la fonction fixe tessellator : les paramètres des points d'échantillonnage, qui sont soit des coordonnées barycentriques triples uvw (domaine tri), ou coordonnées deux dimensions uv (domaines quad et isolignes).

En se basant sur ces entrées, les patches sont évalués, et un vertex de sortie est généré pour chaque échantillon avec ses attributs (tel que les coordonnées de texture et les normales de surfaces) qui doivent être calculées et transmises avec les positions des vertex

émis. Ces vertex sont ensuite triangulés en utilisant la topologie générée par la fonction fixe tessellator et traités par les autres étapes du pipeline graphique. Les nouveaux ajouts de pipeline permettent la tessellation des patches d'entrée avec une densité adaptative. Contrairement à géométrie shaders, le niveau de la tessellation matérielle est nettement plus rapide, plus flexible et permet d'avoir une densité plus élevée dans la tessellation.

4.4.3. Displacement Mapping

En 1978, Blinn a proposé une méthode qui s'appelle le Bump mapping, cette méthode simule les rides dans une surface en perturbant leurs normaux (Blinn, 1978). Bien que cela imite l'ombrage d'une surface de haute résolution, la géométrie elle-même reste inchangée. Cela a conduit Cook (Cook, 1984) à développer le displacement mapping pour donner des objets silhouettes plus réalistes.

Le displacement mapping a été utilisé comme un moyen efficace pour représenter ou animer des objets 3D avec des détails de surface de haute fréquence (Szirmay & Umenhoffer, 2008). Bien que le mappage de texture attribue la couleur aux points de surface, le displacement mapping attribue des décalages, qui sont soit de type scalaire soit vectoriel (voir Figure 4-9).



Figure 4-9 : Comparaison entre le Bump Mapping et le Displacement Mapping

Les avantages du displacement mapping comporte deux volets. Tout d'abord, seuls les sommets d'un maillage de basse fréquence doivent être mis à jour pour animer le

modèle. Deuxièmement, les informations sur la connectivité du maillage de basse fréquence demandent moins d'espace mémoire pour stocker son équivalent en maillage haute fréquence. Le déplacement mapping est réalisé par la tessellation du maillage de base et en déplaçant les sommets générés en fonction de la valeur stockée dans la carte de déplacement (voir Figure 4-10).

La tessellation matérielle est idéalement adaptée au déplacement mapping. Un maillage de basse résolution fournit une surface de base, cette surface est subdivisée sur le GPU pour former un maillage triangulaire dense et immédiatement pixellisé sans utiliser les E/S mémoire (voir Figure 4-11).



Figure 4-10 : Rendu de "Unigine Heaven Benchmark" avec (à droite) et sans (à gauche) l'utilisation de tessellation matérielle.

Note : Bien que la géométrie de base soit la même pour les deux rendus, la géométrie hautement détaillée à droite est obtenue en tessellant d'abord la maille de base puis en utilisant une carte de déplacement.



Figure 4-11 : Exemple de l'utilisation de tessellation matérielle pour le rendu d'une surface avec des petites reliefs.

5. Reconstruction d'un maillage 3D avec vecteurs de déplacements pour les images médicales

5.1. Introduction

Grâce au développement technique des appareils récents utilisés dans le domaine médical, la résolution et le nombre des images médicales obtenues devient plus importants. La reconstruction d'un maillage 3D de l'anatomie humaine à partir de ces images génère un nombre très important de polygones. Préserver tous les détails qu'on a dans les images 2D, conduit à des problèmes au niveau de la quantité des informations stockées et aussi au niveau de la complexité des calculs d'affichage pour avoir une visualisation en temps réel.

Dans ce chapitre, nous introduisons une nouvelle technique de reconstruction d'un maillage 3D à partir d'une séquence de coupes d'images médicales. Cette technique se base sur un maillage de basse résolution (maillage de base) obtenu à l'aide de l'algorithme de Marching Cubes adaptatif et se base aussi sur la carte de déplacement qui contient les détails fins, omis dans le maillage de base. Cette carte est fournie à l'aide des informations extraites à partir des données des images médical 2D.

Notre méthode permet de présenter toutes les informations utiles qu'on a dans les images médicales 2D, dans un maillage haute résolution 3D, avec une optimisation très importante au niveau du stockage et aussi l'accélération des calculs de visualisation par l'utilisation de la carte de déplacement dans les nouvelles unités de tessellation dans les récents GPUs.

5.2. Méthodologie

Dans cette partie du chapitre, nous expliquerons l'algorithme développé pour la reconstruction d'un maillage 3D haute résolution à partir d'une séquence d'images médicales 2D. Au début, nous déterminerons les entrées de notre méthode et nous expliquerons l'idée de base de l'algorithme. Puis, nous décrirons plus en détail les étapes de l'algorithme :

- La modification du maillage 3D basse résolution.

- L'extraction des reliefs de chaque polygone à partir des images 2D.
- La génération automatique de la carte de déplacement.

5.2.1. Vue d'ensemble du nouvel algorithme

Dans notre nouvelle démarche, nous nous basons sur une séquence d'images médicales 2D. Ces images sont soumises à une segmentation afin d'extraire seulement l'organe à reconstruire. Puis nous appliquons sur ces images segmentées l'algorithme de Marching Cubes adaptative (nous devons utiliser les améliorations apportées sur l'algorithme de Marching Cubes, vue dans le chapitre III, pour surpasser les cas d'ambigüité). La taille des cubes choisie dans cet algorithme doit être importante pour obtenir un maillage de basse résolution et en même temps doit être adaptative pour conserver certaines formes qui ne peuvent pas être détectées par ces grands cubes.

L'algorithme développé vise à reconstruire un maillage 3D haute résolution relatif à une anatomie humaine, qui représente toutes les informations utiles que nous avons dans les images médicales. Notre méthode consiste à extraire les détails perdus de chaque triangle du maillage de base à l'aide des informations que nous avons dans ces images. Ces détails sont enregistrés sous forme de vecteurs de déplacement. Et à l'aide de cette méthode nous pouvons aussi générer automatiquement la carte de déplacement pour notre maillage de base (voir Figure 5-1).

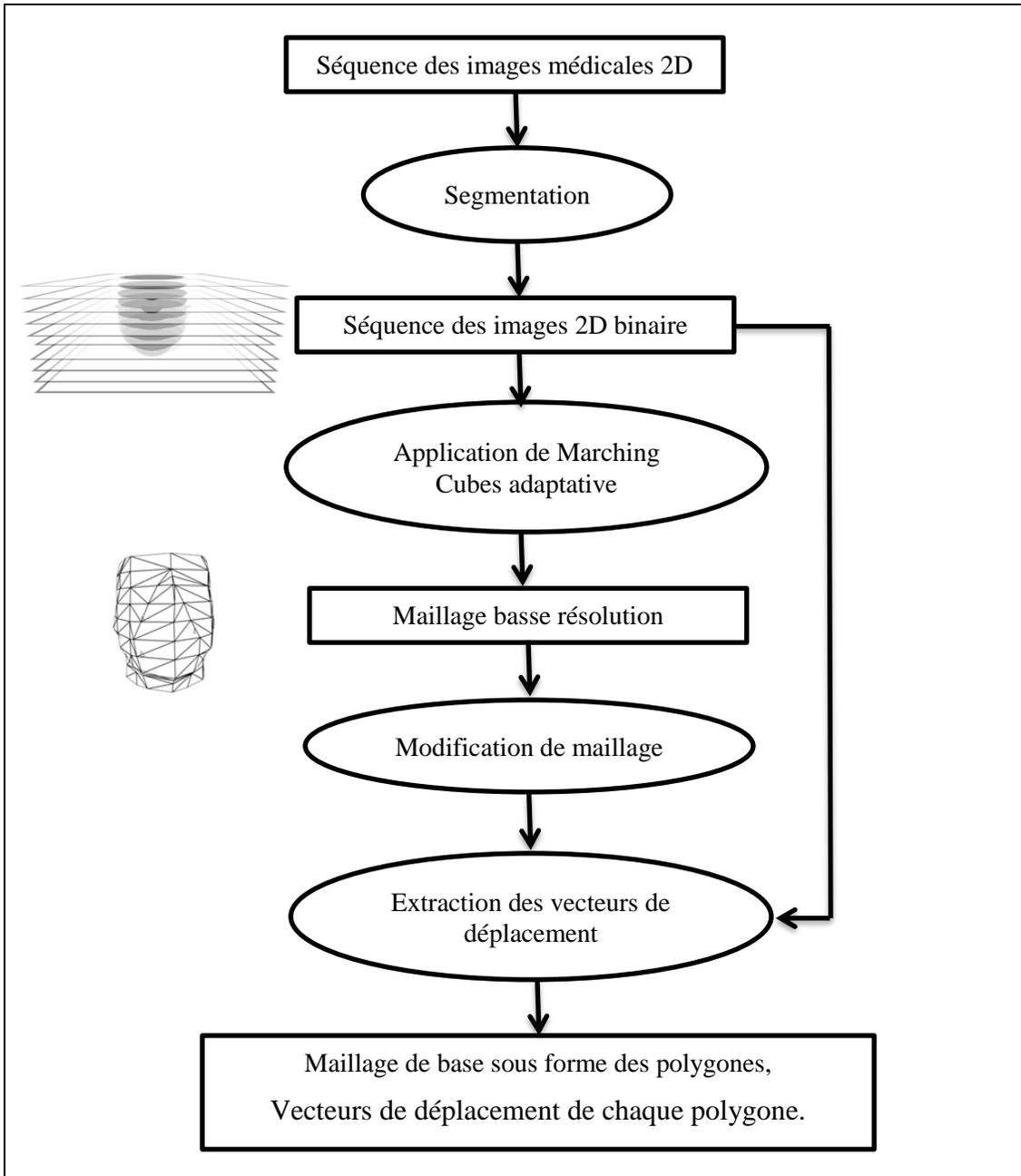


Figure 5-1 : schéma représentant les différentes étapes de notre algorithme.

5.2.2. Application de Marching Cubes Adaptative

L'échantillonnage de l'algorithme de Marching cubes original sur une grille régulière, conduit à des problèmes concernant le traitement des signaux classiques comme le sous-échantillonnage, dans lequel les détails sont perdus, et le sur-échantillonnage, ce qui traduit en général un échantillonnage redondant et le stockage des données inutiles.

Peu de temps après, l'échantillonnage sur les structures adaptatives a été introduit, ce qui entraîne un processus d'échantillonnage plus optimisé.

Ces algorithmes reposent sur des structures de données hiérarchiques pour développer et explorer seules les régions d'intérêt pour l'algorithme. Ces zones d'intérêt pourraient souvent être classées comme des détails de surface. Différents algorithmes prennent des approches différentes pour la subdivision de l'espace (Shu, et al., 1995) (Ju, et al., 2002) (Ho, et al., 2005).

La fonction de la résolution adaptative conduit au problème des fissures dans le résultat final. Elles se produisent lorsque deux cellules adjacentes ont une résolution différente (voir Figure 5.2). Certaines techniques évitent ce problème en intégrant des méthodes différentes.

Une grande variété de ces techniques implique un post-processus connu sous le nom de crack-patching. Il se fait de différentes manières par différents algorithmes. Par exemple, (Shu, et al., 1995), localise et « polygonise » les fissures, qui se traduit par une discontinuité dans la surface. Généralement le crack-patching est un traitement lourd et indésirable.

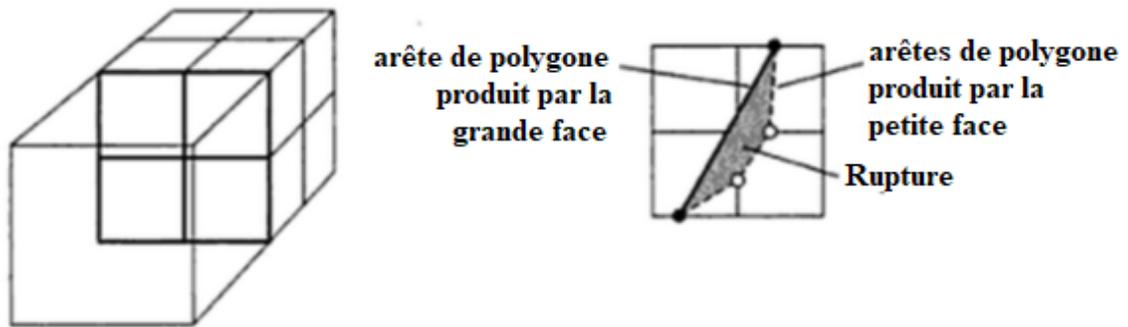


Figure 5-2 : Fissure sur une face commune entre les cellules de subdivision différente (Bloomenthal & Wyvill, 1997).

Dans notre algorithme de reconstruction du maillage basse résolution, nous avons choisis l'application du Marching Cube avec résolution adaptative, sans utiliser des traitements supplémentaires pour le problème des fissures. Ces fissures disparaissent automatiquement après l'intégration des petits reliefs sur le maillage de base (vecteur de déplacement).

5.2.3. Modification du maillage basse résolution

Après l'application de Marching Cubes Adaptative, nous obtenons un maillage basse-résolution sous forme de triangles. Les positions des points des côtés qui existent entre deux images 2D sont estimées par l'application de Marching Cubes, car nous n'avons pas d'informations réelles entre ces deux images 2D (coupes consécutives fournies par l'appareil de saisie médical). Ainsi, lors de l'intégration des « détails perdus » directement dans deux triangles ayant un côté commun (ce côté n'appartient à aucune coupe), nous aurons une déformation dans le résultat final (voir Figure 5-3).

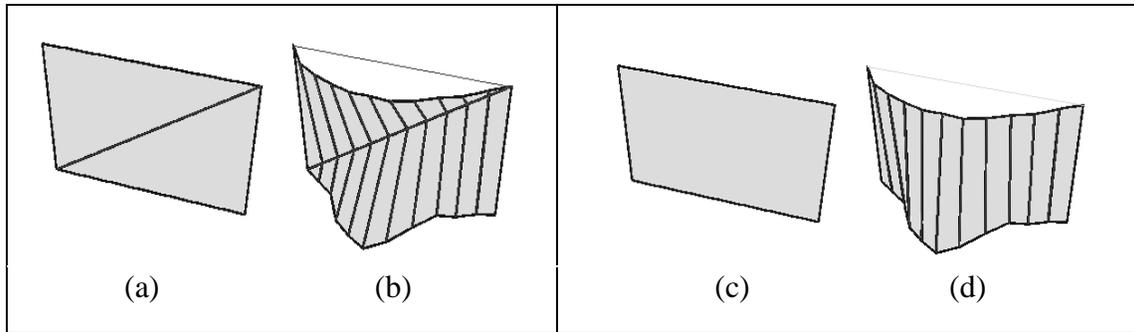


Figure 5-3 : Modification du maillage.

Note : Déformation de maillage (b) après la génération des détails directement dans le maillage de base (a), la suppression de certains côtés (c) améliore l'affichage de l'intégration de ces détails (d).

Le but de la modification du maillage de base est de diminuer le nombre des arêtes entre les images.

Ce traitement est divisé en 2 étapes :

- Dans la 1ère étape, nous supprimons tous les sommets des triangles qui sont entre deux images 2D. En effet, nous testons si un triangle possède un sommet entre deux images et une arête dans l'une, si ces deux conditions sont vérifiées, nous devons faire une triangulation à nouveau des polygones ayant ce sommet afin d'être annulé (voir Figure 5-4). Dans les zones compliquées du maillage, la suppression de certains sommets conduit à une déformation de la surface de base. Pour détecter ces zones, on doit faire le test suivant pour chaque sommet entre deux images successives : les arêtes des triangles ayant ce sommet et qui sont dans ces images doivent être connecté et dans chacune de ces images, nous devons avoir au moins une arête. Si ces conditions ne se sont pas vérifiées, nous n'aurons pas la possibilité de supprimer ce sommet, c'est une zone compliquée.

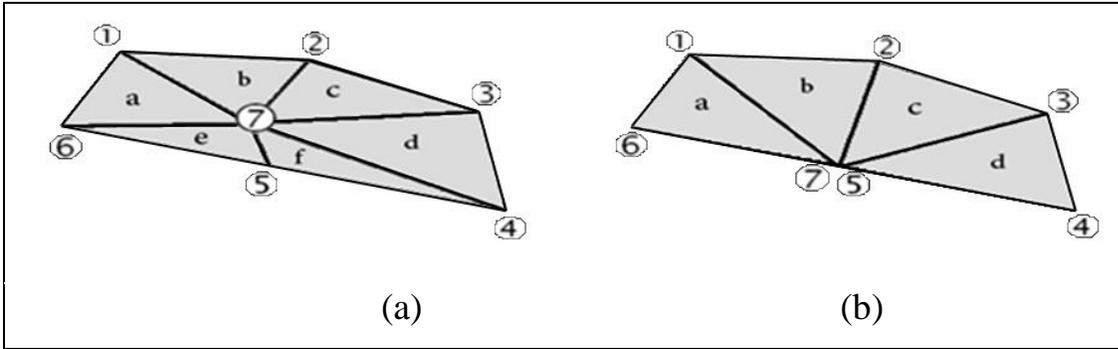


Figure 5-4 : La suppression des sommets entre les images.

Note : Les arêtes $[1,2]$ et $[2,3]$ sont dans l'image n , Les arêtes $[4,5]$ et $[5,6]$ sont dans l'image $n+1$, les autres arêtes sont entre ces deux images. La suppression du point 7 de la forme (a) diminue considérablement les nombres des côtés entre ces images (b).

- Dans la 2ème étape, nous rassemblons les triangles qui ont un côté commun pour obtenir des quadrilatères, ce côté commun doit être entre deux coupes. Dans la plupart des cas, nous trouvons des triangles ayant des côtés communs avec deux autres triangles, donc nous devons choisir le triangle le plus convenable. Dans cette décision nous se basons sur la longueur du côté commun (voir Figure 5-5).

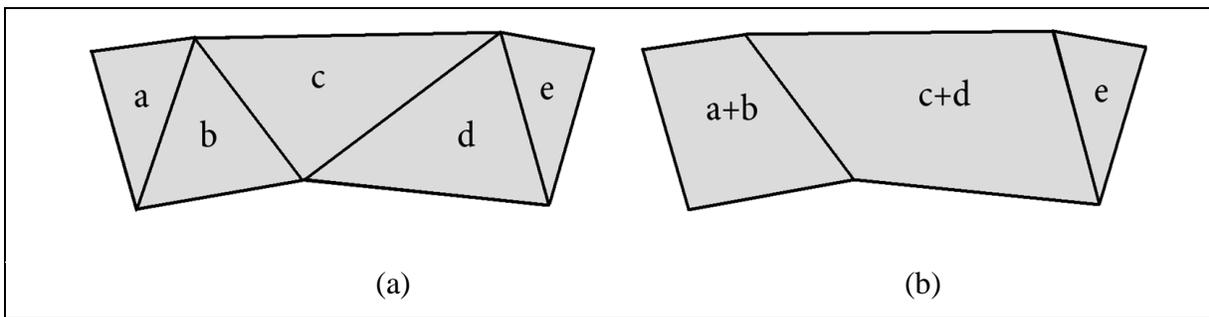


Figure 5-5 : Transformation des polygones triangulaire en quadrilatère.

Note : Assemblage des triangles du maillage de base (a) permet de diminuer le nombre des côtés entre les images (b).

5.2.4. Extraction des vecteurs de déplacement de chaque polygone

L'application de Marching Cubes avec une taille de cube importante donne seulement le maillage de base avec une perte de détails (voir Figure 5-6).

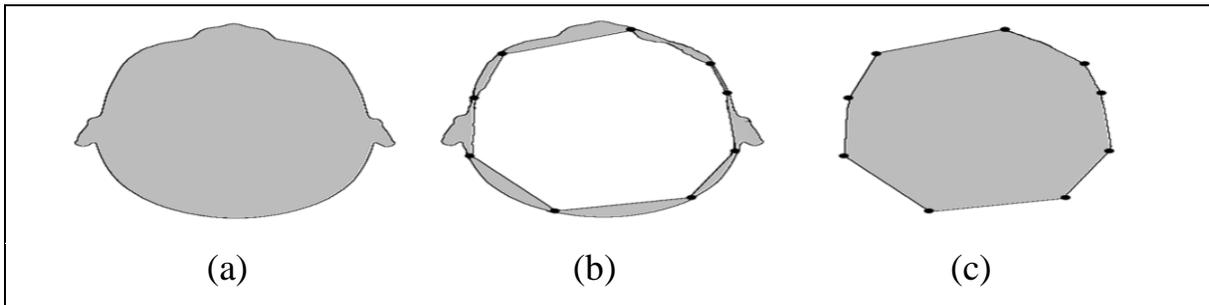


Figure 5-6 : La perte des informations dans le maillage de base.

Note : Après l'application de Marching Cubes sur l'image médicale (a), nous aurons la forme de base (c) et les détails perdu (b).

La forme réelle de l'anatomie humaine qu'on veut reconstruire est souvent sous forme de courbes, par contre le résultat obtenu par MC est basé sur des triangles avec des lignes droites.

Après la modification du maillage de base, on obtient des polygones sous forme de quadrilatères ou de triangles. Pour les côtés existants entre deux images 2D, il n'y a pas de détails à extraire.

Pour chaque côté appartient à une coupe, nous réalisons le traitement suivant :

- a) Nous discrétisons (ou subdivisons) le côté en question en N points (la valeur de N est liée au niveau des détails que nous désirons extraire).
- b) À partir de chaque point de cette discrétisation, nous traçons une ligne perpendiculaire à ce côté porteur afin d'obtenir le point d'intersection avec le contour de l'objet à reconstruire.
- c) Nous déterminons le vecteur entre le point qui appartient au côté et le point d'intersection.

À la fin, en répétant ce traitement à tous les points de discrétisation du côté, nous obtenons pour chacun N vecteurs (voir Figure 5-7).

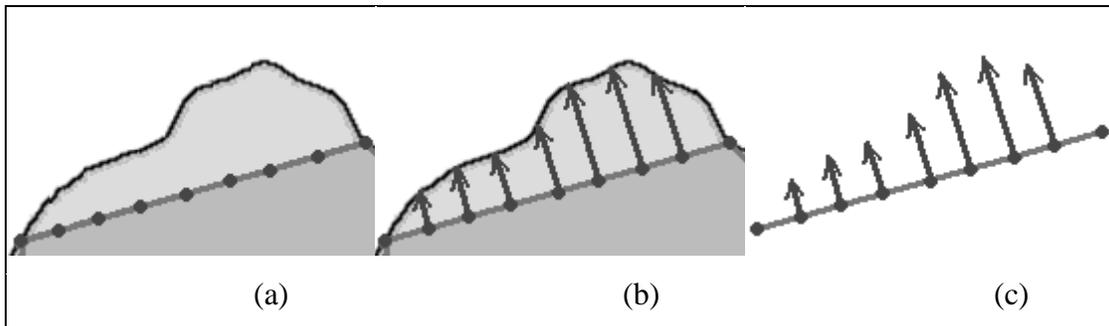


Figure 5-7 : L'extraction des détails perdus.

Note : Chaque côté appartient à une image (a) sera discrétisé en N points pour extraire les vecteurs de déplacement (c).

Les valeurs de ces vecteurs représentent les détails perdus pendant la construction basée sur le maillage de base. Pour chaque vecteur on a trois valeurs à enregistrer x , y et z . Puisque ces vecteurs sont dans les coupes 2D, la valeur de z est constante (l'axe OZ est perpendiculaire à ces coupes), donc dans l'enregistrement de ces vecteurs on se limite aux coordonnées x et y (voir Tableau 5-1).

Structure de base du fichier : M3D v1.0				
Noms		Taille	Description	
En-tête				
	Signature	3 Bytes	'M3D'	
	VertNum (VTN)	4 Bytes	Nombre de vertex du maillage de base.	
	VertSize (VTS)	1 Byte	Détermine le type des coordonnées du vertex.	
	VecSideNum (VCEN)	4 Bytes	Nombre des côtés discrétisé pour extraire les petits reliefs	
	VecSize (VCS)	1 Byte	Détermine le type des coordonnées du vecteur de déplacement.	
	SideDivNum (EDN)	2 Bytes	Nombre de subdivision de chaque côté.	
	FaceNum (FN)	4 Bytes	Nombre de polygones du maillage de base.	
Data				
VertexList	Vertex	Coordonnée X	VTS	Liste des coordonnées de tous les vertex du maillage de base.
		Coordonnée Y	VTS	
		Coordonnée Z	VTS	
	Répéter «VTN» times			
VectorSideList	VectorSide	Coordonnée X	VCS	Liste des côtés subdivisés avec leurs vecteurs de déplacement.
		Coordonnée Y	VCS	
	Repeated «EDN» fois			
	Répéter «VCEN» fois			
FacelList	Face	Index vertex 1	4 Bytes	Liste des polygones de maillage de base, chaque polygone possède quatre vertex et les vecteurs de déplacement de deux côtés qui se trouvent dans les images médicales. Pour les polygones triangulaires : Index vertex 3 = Index vertex 4.
		Index vertex 2	4 Bytes	
		Index vertex 3	4 Bytes	
		Index vertex 4	4 Bytes	
		Index VectorSide 1	4 Bytes	
		Index VectorSide 2	4 Bytes	
Répéter «FN» fois				

Tableau 5-1 : Structure adoptée pour le stockage des vecteurs de déplacement et le maillage de base.

5.2.5. Génération automatique de la carte de déplacement

Le « déplacement mapping » est une technique qui permet d'ajouter des détails à petite échelle sur les surfaces. Avec cette méthode, on peut changer la position géométrique des points de la surface selon la valeur affectée à la carte de déplacement (voir § 4.4.3).

Le déplacement mapping traditionnel utilise une image niveaux de gris « height map » qui indique la distance de déplacement de chaque point de la surface sur une normale fixe. Actuellement on a un autre type de déplacement mapping qui utilise une image RGB « vector map ». Cette image contient des vecteurs 3D qui permettent de déterminer la distance et la direction qu'on utilise pour déplacer les points de la surface.

La génération de la carte de déplacement de notre maillage basse résolution est basée sur les détails extraits dans la section précédente. Pour chaque polygone quadrilatère, on a extrait les détails de ses deux côtés à partir des images médicales. Les autres points de la surface de ce polygone sont calculés à l'aide de ces deux côtés. Après l'intégration des détails dans chacun de ces deux côtés on aura deux courbes $V1(u)$ et $V2(u)$. La détermination de la position des autres points de cette surface est basée sur la formule de la surface réglée (voir Figure 5-8), cette surface est créée à l'aide des deux courbes $V1(u)$ et $V2(u)$ dont laquelle on applique la relation suivante :

$$Q(u, v) = V1(u) + v \times (V2(u) - V1(u)) \text{ Tel que } u \text{ et } v \text{ entre } 0 \text{ et } 1.$$

Les polygones triangulaires sont des cas particuliers des polygones quadrilatère où $V2$ (ou $V1$) est constant.

La structure de la carte de déplacement, qui montre les petits reliefs du maillage de base, est présentée dans la Figure 5-9.

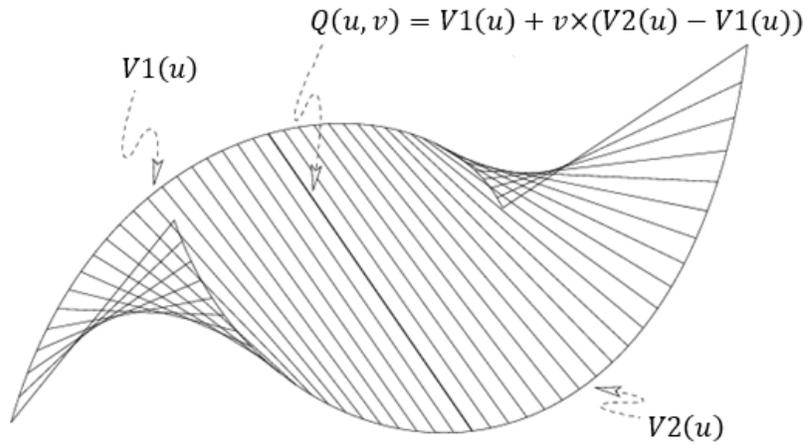


Figure 5-8 : L'interpolation des points de surface à l'aide de deux courbes $V1(u)$ et $V2(u)$.

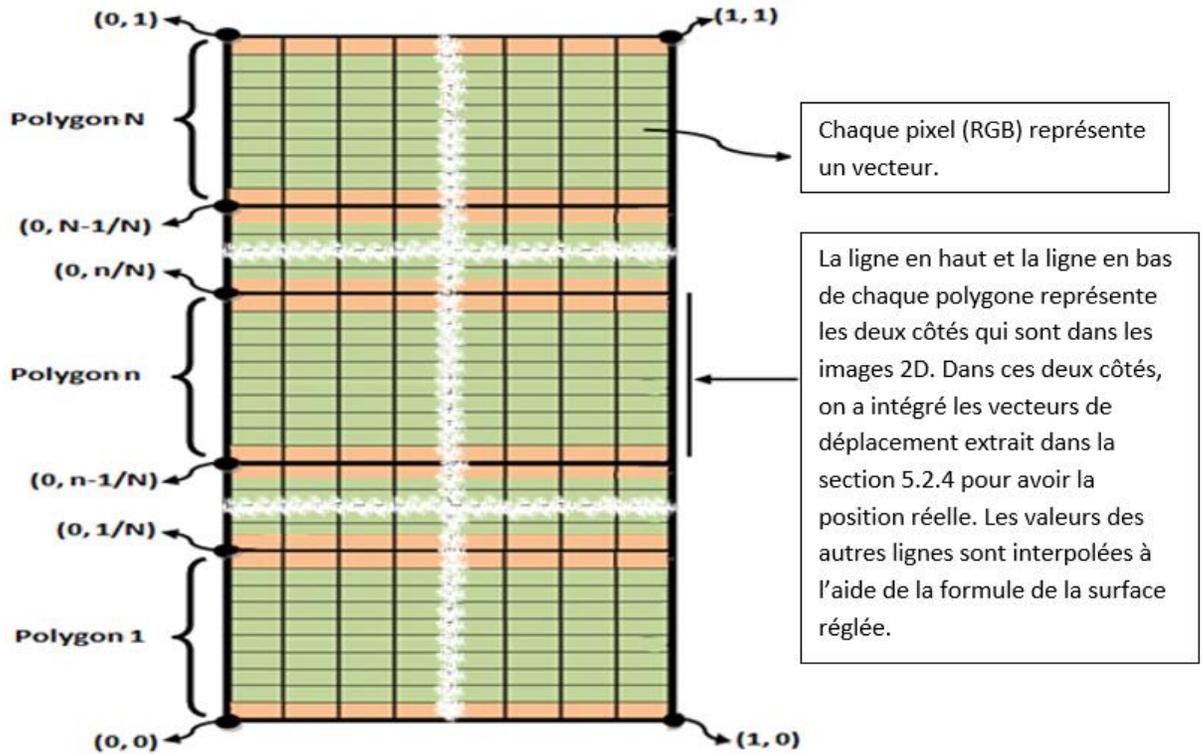


Figure 5-9 : Structure de la carte de déplacement.

5.3. Résultats & Discussions

Dans la première étape, nous avons testé notre algorithme au niveau de la qualité du maillage 3D finale obtenue. L'utilisation de MC pour construire un maillage haute résolution oblige de choisir une taille très petite des cubes utilisés. Ce choix conduit à avoir des surfaces ayant un aspect de « marches d'escalier », surtout lorsque la distance entre les images est importante (voir Figure 5-10). Et cela demande d'ajouter des traitements supplémentaires pour lisser le maillage déformé, ce qui pourra changer la position de certains vertex. Par contre, avec notre algorithme, on n'aura pas ce problème puisque le maillage de base est obtenu par l'application de MC avec des cubes ayant une grande taille, puis on ajoute les détails perdus (voir Figure 5-11).

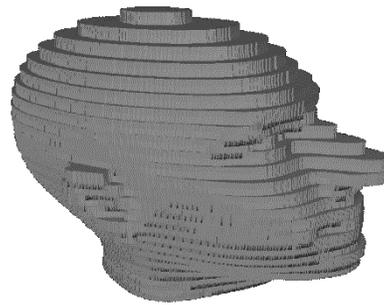


Figure 5-10 : l'aspect de « marches d'escalier » sur les surfaces obtenu par le Marching Cubes.

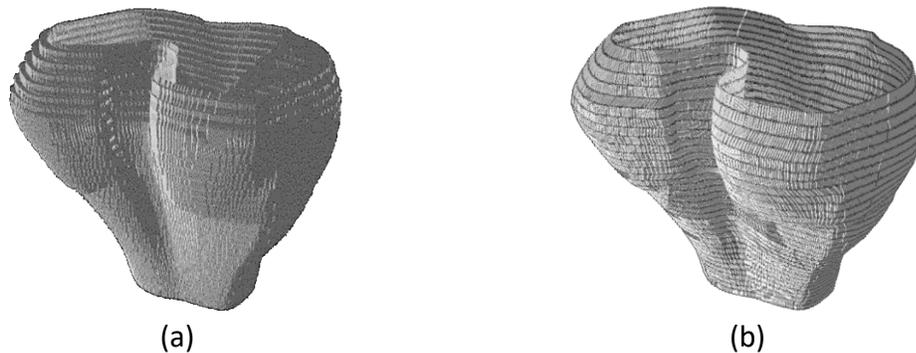


Figure 5-11 : Comparaison de la qualité du maillage obtenu avec notre méthode (b) et le maillage de Marching Cubes (a).

Dans la deuxième étape, nous avons testé notre algorithme au niveau du stockage, nous comparons la quantité des informations stockées en utilisant la structure que nous avons adoptée dans la section 5.2.4, avec la structure la plus compacte pour le stockage du maillage que nous avons vu dans la section 3.3.1.

Pour simplifier les calculs, nous avons utilisé pour le maillage de base un cylindre, nous aurons les mêmes résultats pour les maillages complexe car il suffit de déplacer les vertex du cylindre pour obtenir n'importe quelle forme 3D.

Le nombre de subdivisions horizontales et verticales du cylindre que nous avons utilisé dans cette comparaison est 10 (voir Figure 5-12 (a)). Avec ce maillage, nous aurons les informations suivantes :

- a) Nombre de vertex de base = 100,
- b) Nombre de polygone triangulaire de base = 180,
- c) Nombre des segments à diviser pour ajouter les détails = 91,

Dans la comparaison nous avons pris plusieurs valeurs pour le nombre des vecteurs à utiliser pour chaque segment divisé.

Pour ajouter des détails à ce cylindre de base, nous devons appliquer la tessellation, cette opération augmente le nombre de vertex et polygone selon le nombre des vecteurs utilisé pour chaque segment divisé (voir Figure 5-12 (b)).

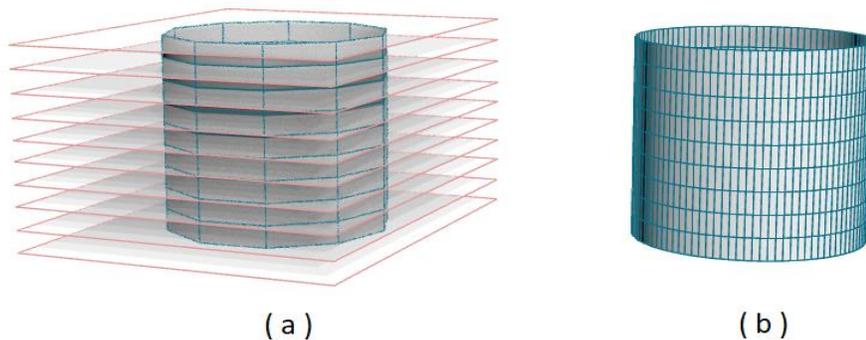


Figure 5-12 : Le maillage tesselé (b) obtenu à partir le maillage de base (a).

Le nombre de vertex (NV) obtenu après la tessellation est calculé par la formule suivante :

$$NV = NVB + NSD \times ND$$

NVB : Nombre de vertex de base

NSD : Nombre des segments à diviser

ND : Nombre de division de chaque segment.

Pour le nombre de polygone (NP), nous avons la formule suivante :

$$NP = NPB \times (ND + 1)$$

NPB : Nombre de polygone de base

ND : Nombre de division de chaque segment à diviser

Dans cette comparaison, nous avons supposé que les coordonnées des vertex sont de type « float », les index des vertex sont de type entier et les coordonnées des vecteurs de déplacement sont codées sur un octet.

Pour les vecteurs de déplacement, cette taille est suffisante, car les valeurs des coordonnées ne peuvent pas dépasser la taille de la diagonale de la base du cube utilisé dans le Marching Cubes. Avec un octet nous avons la possibilité d'enregistrer des valeurs entières entre -127 et 128. D'un autre côté, la taille de diagonale de la base du cube est généralement inférieure à 127 sauf dans des cas particuliers où la résolution des images médicales 2D est très grande. Dans ce cas les coordonnées de ces vecteurs seront enregistrées dans un espace de 2 Octets ou de choisir un maillage de base plus petit.

En fin, nous estimons la taille des informations à stocker pour le maillage final obtenu après la tessellation.

La méthode classique pour le stockage du maillage se base sur deux listes. La première liste contient les coordonnées des vertex, la deuxième liste contient les indices des vertex de chaque polygone.

Pour chaque vertex de la première liste, nous avons 3 coordonnées (x, y et z) de type float (4 Octets), et pour chaque triangle de la deuxième liste, nous avons 3 indices de type entier (4 Octets). Donc, La taille totale de ces deux listes se calcule de la manière suivante :

$$\text{StockageClassique} = 12 \times \text{NV} + 12 \times \text{NP}$$

NV : Le nombre de vertex après la tessellation

NP : Le nombre de polygones après la tessellation

Notre nouvelle méthode de stockage est basée sur la structure que nous avons créée dans la section 5.2.4. Les informations du maillage final se divisent en deux parties : maillage de base et vecteurs de déplacement.

Le maillage de base est stocké avec la méthode classique ; une liste des vertex de base et une autre liste pour les indices des polygones quadrilatères. Pour chaque polygone, nous aurons 6 indices : 4 indices pour indiquer la position des vertex du quadrilatère et 2 autres indices pour indiquer la position des vecteurs de déplacement de deux arêtes se situant dans les coupes **n** et **n+1** (voir Figure 5-13).

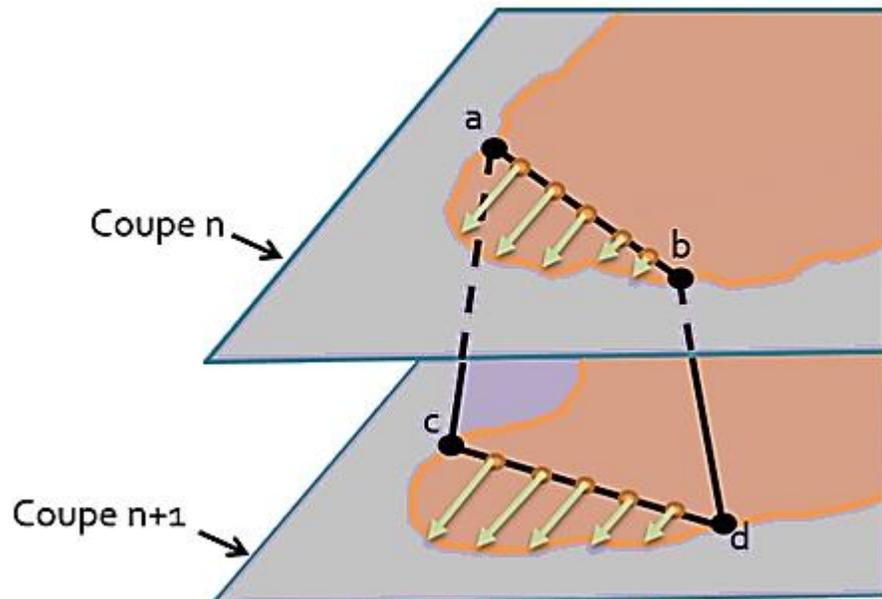


Figure 5-13 : Représentation d'un polygone quadrilatère [abcd] du maillage de base avec ces vecteurs de déplacement.

Note : le polygone [abcd] possède 4 indices de ces vertex et 2 autres indices concernant les vecteurs de déplacement ; un indice qui indique les vecteurs de déplacement de l'arête [ab] et autre pour indiquer les vecteurs de l'arête [cd].

Les vecteurs de déplacement sont stockés dans une liste. Chaque élément de cette liste contient les vecteurs de déplacement d'une arête appartenant à une coupe.

La formule finale qui regroupe le maillage de base et les vecteurs de déplacement pour le calcul de la taille totale est la suivante :

$$\text{StockageNouvel} = (12 \times \text{NVB} + 24 \times \text{NPQB}) + \text{NSD} \times \text{NV} \times 2.$$

NVB : le nombre de vertex de base

NPQB : le nombre de polygone quadrilatère du maillage de base

NSD : le nombre des segments à divisé

NV : le nombre des vecteurs de chaque arête

Sachant que le nombre de polygone de base quadrilatère est égal à la moitié des polygones de base triangulaire.

Dans le Tableau 5-2, nous avons comparé notre méthode de stockage des informations concernant le maillage haute résolution avec la méthode classique.

A partir des résultats obtenus, nous remarquons que notre méthode permet d'avoir une optimisation très importante dans le stockage par rapport la méthode classique.

Nombre des vecteurs calculé pour chaque segment (N)	Maillage haute résolution		Stockage classique du maillage haute résolution (Octet)	Stockage avec notre méthode du maillage haute résolution (Octet)	Facteur d'efficacité de stockage (nouvelle approche / classique)
	Nombre de vertex	Nombre de polygone			
1	191	360	6612	3542	53,57%
3	373	720	13116	3906	29,78%
5	555	1080	19620	4270	21,76%
10	1010	1980	35880	5180	14,44%
20	1920	3780	68400	7000	10,23%
30	2830	5580	100920	8820	8,74%
40	3740	7380	133440	10640	7,97%
50	4650	9180	165960	12460	7,51%

Tableau 5-2 : Comparaison de stockage entre notre nouvelle méthode et la méthode classique.

Finalement, avec notre méthode nous pouvons générer la carte de déplacement pour le maillage de base d'une manière automatique (voir §5.2.5). La génération de cette carte a pour but d'être exploitable par le displacement mapping (voir §4.4.3). Nous avons vu dans le chapitre précédent que l'utilisation du displacement mapping avec la nouvelle architecture de carte graphique, surtout avec l'unité de tessellation matérielle, permet d'accélérer les calculs de rendu.

Pour comparer le temps de rendu, nous avons fait plusieurs tests sur des maillages avec différentes tailles. Et chaque fois, nous enregistrons le nombre des images par seconde (fps) obtenu par le rendu classique et le rendu avec tessellation matérielles (voir Figure 5-14).

A partir des résultats obtenus, nous remarquons que le rendu avec tessellation matérielles accélère les calculs d'affichage jusqu'à 26 fois plus par rapport au rendu classique (voir Tableau 5-3).

Nombre des polygones de maillage final	Nombre des polygones de maillage de base	Rendu classique (fps)	Rendu avec tessellation matérielle (fps)	Facteur d'efficacité de rendu (avec tessellation/classique)
128	128	1260	1130	0,90
512	128	1200	1150	0,96
2048	128	705	1126	1,60
8192	128	248	1150	4,64
32768	128	68	870	12,79
131072	128	18,6	243	13,06
524288	128	4,6	91	19,78
1048576	256	2,3	49	21,30
2097152	512	1,1	26	23,64
3145728	768	0,77	18	23,38
4194304	1024	0,59	14	23,73
5242880	1280	0,44	11,1	25,23
6291456	1536	0,39	9,6	24,62
7340032	1792	0,32	8,3	25,94
8388608	2048	0,29	7,3	25,17
12582912	2176	0,19	5	26,32

Tableau 5-3 : comparaison entre le temps de rendu classique et le rendu avec tessellation matérielle

Note : Configuration matérielle utilisé :

- GPU : Radeon R5 (128 unité de calcul Shader à une fréquence de 800 MHz)
- CPU : AMD A8-6410 (4 cœurs à une fréquence de 2GHz)

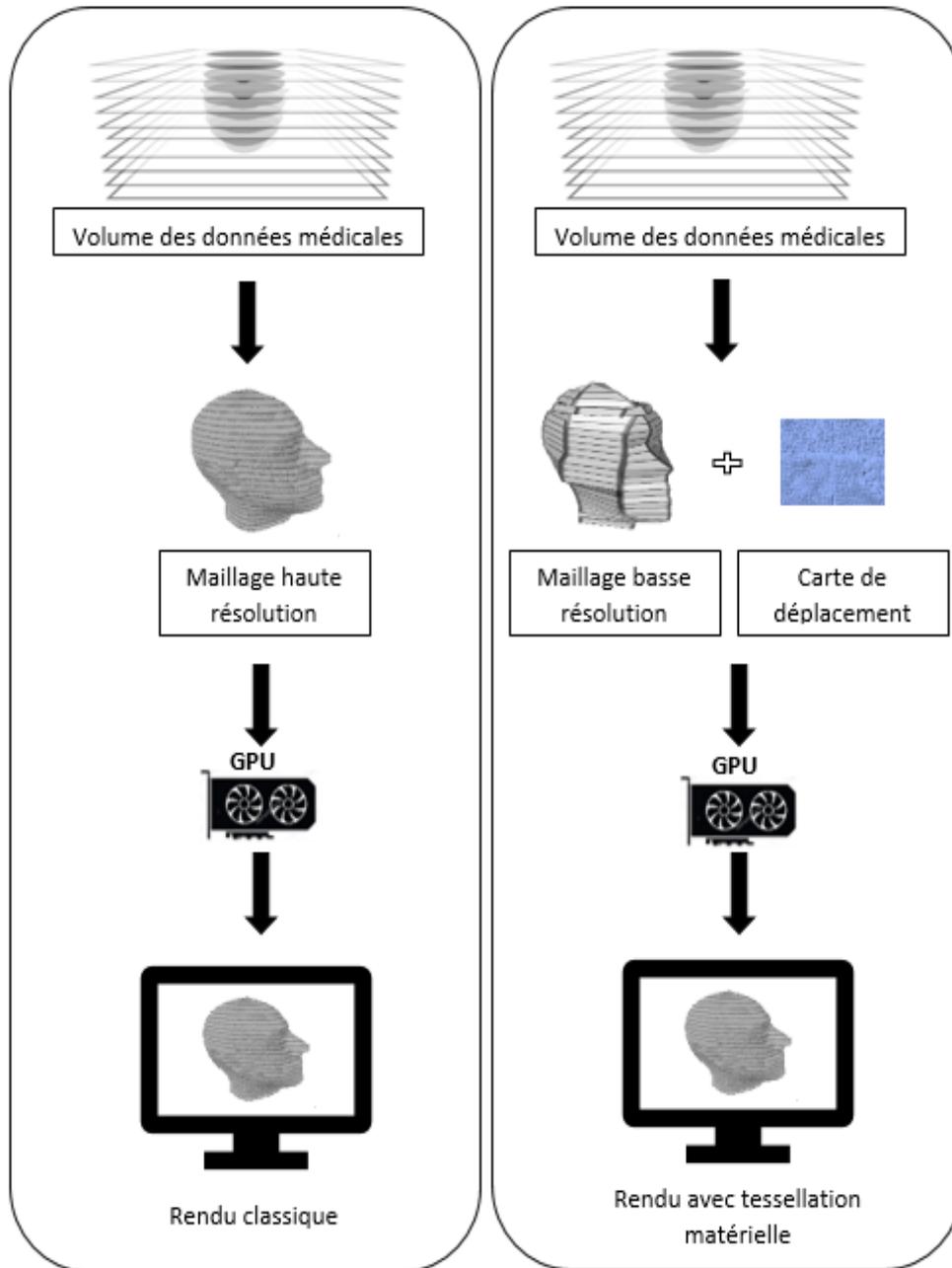


Figure 5-14 : Le rendu classique et le rendu avec tessellation matérielle.

5.4. Conclusion

La qualité du maillage 3D obtenu à l'aide de Marching Cubes dépend de la taille de cube utilisé dans cet algorithme. Dans le domaine médical, la taille des cubes choisie doit être petite pour conserver toutes les informations que nous avons dans les coupes 2D. Ce choix donne un maillage de haute résolution mais très lourd au niveau du stockage et d'affichage.

Dans notre travail, nous avons appliqué l'algorithme de Marching Cubes sur les images médicales. La taille des cubes utilisés dans cet algorithme est importante à fin d'obtenir un maillage basse résolution. Les détails perdus dans la reconstruction de ce maillage sont régénérés en se basant sur les informations que nous avons dans les images médicales. Le maillage de base est modifié d'une manière qui permet d'intégrer les détails perdus directement sur les polygones du maillage.

Notre approche a prouvé son intérêt en deux points principaux:

- a) La représentation des petits reliefs du maillage de base sous forme de vecteurs 2D conduit à une optimisation très importante dans le stockage final du maillage haute résolution.
- b) L'extraction des petits reliefs de chaque polygone conduit à la génération automatique de la carte de déplacement. L'utilisation de cette carte avec les GPU récent permet d'accélérer les calculs de rendu.

Cependant, malgré les innovations qu'elle contient, cette approche présente un certain nombre de faiblesses dans sa mise en œuvre et dans certaines situations. Ce thème ainsi que ses solutions font l'objet d'une étude particulière que nous présentons dans le chapitre suivant.

6. Amélioration de l'extraction des vecteurs de déplacement pour un maillage 3D des images médicales

6.1. Introduction

Comme nous l'avons signalé dans les chapitres précédents, à l'aide de l'imagerie médicale, il est possible aujourd'hui, de visualiser l'anatomie interne du patient sans recourir à la chirurgie. Les images acquises par les différentes modalités, sont sous forme de coupes 2D sur lesquelles il n'est pas toujours facile d'interpréter les différents problèmes rencontrés. Pour surmonter ces problèmes, l'imagerie 3D a été développée. Cela permet de mieux représenter l'anatomie interne afin de maximiser les chances de bons diagnostics et, en tout cas en élargissant le public des gens en mesure de faire certaines lectures, plus particulièrement dans la planification chirurgicale : Shiozawa et al. (Shiozawa, et al., 2009) ont montré les avantages de la visualisation en 3D pour la planification du traitement des tumeurs surrénales. De même, (Lee, et al., 2003) ont étudié l'intérêt de l'affichage en 3D des données TDM dans les discussions préopératoire de la chirurgie laparoscopique du cancer gastrique.

Dans le chapitre précédent, nous avons développé une méthode qui permet de créer un maillage 3D à partir d'une série des images médicales. Cette méthode a prouvé son intérêt au niveau de l'optimisation de la quantité des informations à stocker pour le maillage et aussi d'accélérer les calculs d'affichage en utilisant les GPUs récents.

Cette méthode permet d'extraire les détails perdus de chaque polygone du maillage de base à l'aide des informations qu'on a dans les images médicales (saisies au départ). Ces détails sont enregistrés sous forme des vecteurs de déplacement. La génération des vecteurs de déplacement se base sur la discrétisation des côtés de maillage de base existant sur les images médicales 2D en N point, puis nous déterminons les vecteurs de déplacement qui correspond à chaque point.

L'algorithme utilisé pour extraire ces vecteurs souffre de deux problèmes essentiels au niveau de la précision du maillage final haute résolution :

- a) Tous les côtés des polygones sont discrétisés en N point (la valeur de N est fixe), par contre les largeurs de ces côtés ne sont pas égales (voir Figure 6-1), donc le niveau des détails à extraire pour chaque polygone ne sera pas

le même. Le maillage 3D final ne sera pas homogène au niveau des détails extraits de chaque polygone.

- b) Dans certaines situations, nous avons des erreurs au niveau de la forme finale de l'anatomie à reconstruire. Le traçage d'une ligne perpendiculaire à partir des points obtenus après la discrétisation de chaque côté, donne parfois deux points d'intersection ou plus avec le contour qu'on a dans les images 2D. Avec la méthode d'extraction de l'algorithme d'origine, nous choisissons toujours le vecteur que nous avons entre le point qui appartient au côté et le point d'intersection la plus proche de ce côté. Ce choix conduit parfois à extraire des détails qui ne correspondent pas à la forme réelle (voir Figure 6-2).

Dans ce chapitre, nous proposons une nouvelle méthode d'extraction de vecteurs de déplacement pour dépasser les problèmes cités précédemment, cette méthode se base essentiellement sur la génération de ces vecteurs par l'extraction du contour de l'objet à reconstruire.

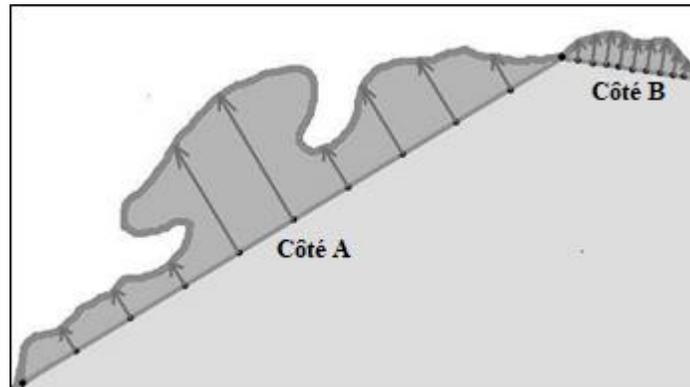


Figure 6-1 : hétérogénéité des détails extraits de chaque côté.

Note : Le niveau des détails extraits du côté B est plus important que celui du côté A.

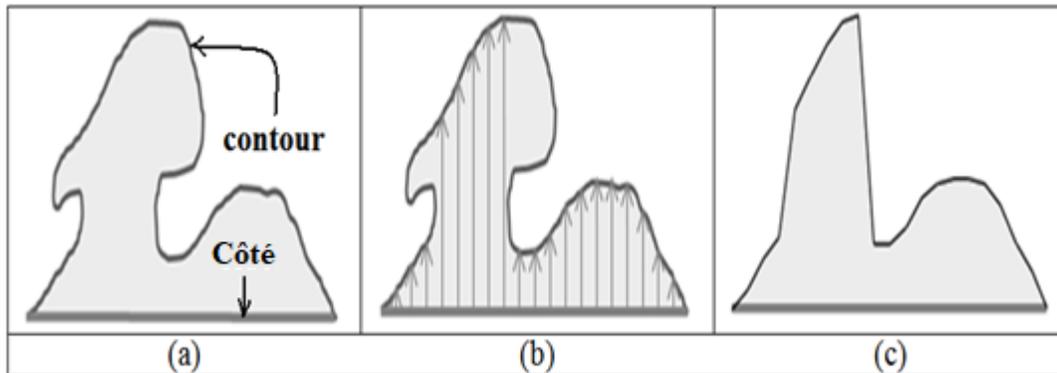


Figure 6-2 : Erreurs d'extraction des petits reliefs.

Note : La forme du contour obtenue (c) à l'aide de vecteurs de déplacement (b) ne correspond pas exactement avec la forme réelle (a).

6.2. Méthodologie

Dans cette partie de chapitre, nous décrivons l'algorithme développé pour extraire les vecteurs de déplacement à partir d'une séquence des images médicales 2D. Au début, nous déterminerons les entrées de notre méthode et nous expliquerons l'idée de base de l'algorithme. Puis, nous décrivons plus en détail les étapes de l'algorithme : L'extraction et la discrétisation des points de contour.

6.2.1. Vue d'ensemble de l'algorithme d'extraction

Pour dépasser les problèmes cités précédemment, on a choisi une nouvelle stratégie pour extraire les vecteurs de déplacement. Cette stratégie se base sur l'extraction du contour de l'objet à reconstruire à partir des images médicales, puis à l'aide de ces contours on génère les vecteurs de déplacement.

Il existe plusieurs méthodes concernant la détection de contour, ces méthodes peuvent être regroupées en plusieurs classes :

- a) Filtrage passe-haut telles que les détecteurs de Prewitt, Sobel et Canny (Canny, 1986).
- b) Analyse multi-échelle développées avec la théorie des ondelettes (Yi, et al., 2009) (Zhang & Bao, 2002).
- c) Filtrage non-linéaire telles que le filtre médian et plus récemment (Laligant & Truchetet, 2010).
- d) Approximations parcimonieuses par dictionnaire redondant (Mairal, et al., 2008).

La détection de contours consiste à localiser les changements abrupts d'intensité des pixels, mais dans notre situation, nous voulons avoir les points du contour dans une liste chaînée, avec un ordre bien déterminé. Donc, nous devons créer un algorithme qui répond à nos besoins.

Les entrées de notre nouvelle méthode d'extraction des vecteurs de déplacement sont les mêmes entrées de l'algorithme origine. Nous avons le maillage de base, Les images médicales qui subissent une segmentation (voir Figure 6-3), et le niveau de détails à extraire en déterminant le pas de discrétisation.

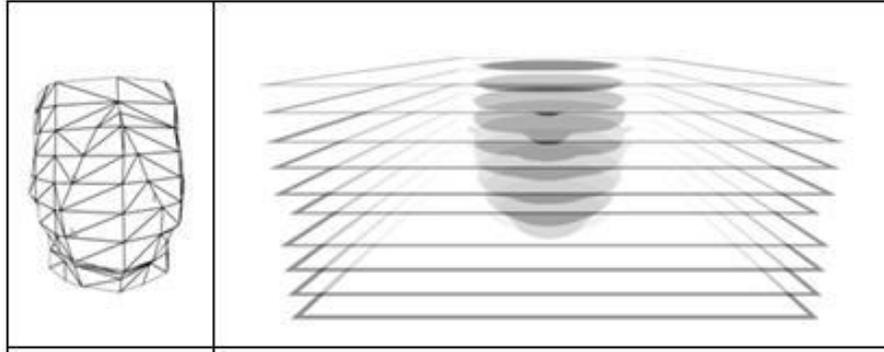


Figure 6-3 : Les entrées de l'algorithme d'extraction des vecteurs de déplacement : Maillage de base (a), images médicales segmentées (b).

La démarche suivie pour l'extraction de contours est la suivante :

Pour chaque arête du maillage de base situant sur le même plan d'une coupe, nous devons extraire le contour de l'objet à reconstruire existant entre ses bornes. Puis, nous discrétisons ces contours selon le niveau de détails déterminé par les paramètres d'entrée. Et finalement, nous déduisons les vecteurs de déplacement qui permet de déterminer les positions des points des arêtes discrétisés.

6.2.2. Extraction du contour

L'extraction du contour s'applique seulement sur les arêtes situées sur le même plan des images 2D. Pour les autres arêtes, il n'y a pas de détails à extraire.

Dans notre algorithme d'extraction du contour on utilise deux listes de points, la liste principale **listePtContour** où nous enregistrons les points du contour et une liste provisoire **listeVide** utilisé pour traiter les cas d'ambiguïté qu'on peut avoir au cours de l'extraction du contour.

Plus précisément, le pseudo-code ci-dessous de l'algorithme 1 et 2 décrit notre algorithme d'extraction du contour.

Algorithme 1 : Extraction des vecteurs de déplacement.

Algorithme ExtractionVectorsDisplacement (MaillageBase, Images2D)**Début**

Pour chaque arête [P1, P2] du maillage de base situé sur le plan d'une image 2D.

{ListePtContour, listeVide} \leftarrow InitialiseListes (P1)

ListePtContour : cette liste enregistre les points du contour entre P1 et P2 de l'objet à reconstruire, le premier élément ajouté à cette liste est le point P1 : (P1.x, P1.y)

ListeVide est initialisée par les points vides voisinage du point (P1.x, P1.y), [point vide : c'est un point (pixel) dans l'image médicale 2D qui n'appartient pas à l'objet à reconstruire]

Faire

Pnext \leftarrow ExtractNextContourPoint (listePtContour,
listeVide, image2D)

listPtContour.add(Pnext)

Tant Que (Pnext < > P2)

Fin Pour**Fin**

***Note** : Compte tenu de maillage de base et les images médicales segmentées, cette procédure extrait les vecteurs de déplacements de chaque côté du maillage de base existant dans les images médicales.*

Algorithme 2 : Extraction du point du contour suivant.

Algorithme ExtractNextContourPoint(listePtContour, listeVide, Image2D)

Début

$P \leftarrow$ Dernier élément ajouté à listePtContour

ListeVide \leftarrow DetermineVoisinageVideValide (P)

Tous les points vides voisinage valides de P doivent être connectés et liés aux points qui existent dans listeVide.

listePoints \leftarrow DetermineListePointContourProbable (P)

Dans cette fonction (DetermineListePointContourProbable), nous testons les 4 directions possible (haut, bas, gauche et droite) autour de P pour choisir le point du contour suivant, les conditions à vérifier pour être le point suivant probable du contour sont :

- Doit être dans l'objet à reconstruire.
- Avoir un point voisinage vide lié au point de listeVide.

Psuivant \leftarrow SelectionPointSuivantContour (listePoints)

Parmi les points probables d'être le point du contour suivant, nous choisissons le point qui n'existe pas dans listePtContour. Si tous les points probable existent dans cette liste, nous choisissons le premier point ajouter à listePtContour.

Retourn (Psuivant)

Fin

Note : L'algorithme permet de retourner le point du contour suivant par rapport le dernier point ajouté à la liste listePtContour.

6.2.3. Discrétisation des points du contour

Après avoir extrait les points du contour, ce dernier doit être discrétisé selon le niveau de détail que nous voulons avoir lors de la visualisation de l'objet 3D.

Dans notre algorithme, nous avons utilisé le paramètre **PasDisc** pour déterminer le pas de discrétisation. Avec $\text{PasDisc} = k$, signifie que chaque k points du contour est remplacé par un vecteur de déplacement, à la fin de ce traitement on obtient pour chaque côté n vecteur de déplacement, la valeur de n change selon le nombre de points de contours (voir algorithme 3). Pour stocker tous les points du contour, le pas de discrétisation doit avoir la valeur 1.

Algorithme 3 : Discrétisation du contour d'une arête

Algorithme DiscrétisationContour (listePtContour, PasDisc)

Début

$n \leftarrow \text{listePtContour.count} / \text{PasDisc}$

n représente le nombre de vecteurs de déplacement obtenus après la discrétisation de ce contour.

Pour $i \leftarrow 0$ à n

$\text{listeVec.add}(\text{listePtContour}((i+1)*\text{PasDisc}) - \text{listePtContour}(i*\text{PasDisc}))$

listeVec enregistre à la fin de ce traitement, la liste des vecteurs de déplacement à stocker pour une arête.

$i \leftarrow i+1$

Fin Pour

Fin

Structure de base du fichier : M3D v2.0					
		Noms	Taille	Description	
En-tête					
	Signature		3 Bytes	'M3D'	
	VertNum (VTN)		4 Bytes	Nombre de vertex du maillage de base.	
	VertSize (VTS)		1 Byte	Détermine le type des coordonnées du vertex.	
	VecSideNum (VCEN)		4 Bytes	Nombre de côtés discrétisés pour extraire les petits reliefs	
	VecSize (VCS)		1 Byte	Détermine le type de coordonnées du vecteur de déplacement.	
	FaceNum (FN)		4 Bytes	Nombre de polygones du maillage de base.	
Data					
Vertexlist	Vertex	Coordonnée X	VTS	Liste des coordonnées de tous les vertex du maillage de base.	
		Coordonnée Y	VTS		
		Coordonnée Z	VTS		
	Répéter «VTN» fois				
VectorSideList	VectorSide	NumVectors (NV)		Liste des côtés subdivisés avec leurs vecteurs de déplacement.	
		Vector	Coordonnée X		VCS
			Coordonnée Y		VCS
	Répéter «NV» fois				
	Répéter «VCEN» fois				
Facelist	Face	Index vertex 1	4 Bytes	Liste des polygones de maillage de base, chaque polygone possède quatre vertex et les vecteurs de déplacement de deux côtés qui se trouvent dans les images médicales. Pour les polygones triangulaires : Index vertex 3 = Index vertex 4.	
		Index vertex 2	4 Bytes		
		Index vertex 3	4 Bytes		
		Index vertex 4	4 Bytes		
		Index VectorSide 1	4 Bytes		
		Index VectorSide 2	4 Bytes		
	Répéter «FN» fois				

Tableau 6-1 : Structure adoptée pour le stockage des vecteurs de déplacement extrait à partir du contour et le maillage de base.

6.3. Résultats & Discussions

Dans la 1ère étape, nous avons testé la nouvelle version d'extraction des petits reliefs au niveau de la qualité du maillage 3D final obtenu.

La méthode d'extraction utilisée dans sa première version, se base sur la discrétisation des arêtes qui sont sur les images 2D en N points, puis d'extraire pour chaque point le vecteur de déplacement correspondant, la valeur de N est fixe pour toutes les arêtes, par contre les longueurs de ces arêtes ne sont pas fixes, ce qui ne permet pas de déterminer le même niveau d'extraction dans tout le maillage 3D.

Dans la deuxième version de l'algorithme, l'extraction puis la discrétisation du contour de l'anatomie permet d'avoir le même niveau de détails pour tous les triangles. Pour les petits polygones du maillage de base, généralement le nombre des points du contour est inférieur à celui des autres polygones, de sorte que le nombre de vecteurs de déplacement sera également inférieur à celui des autres polygones. Et vice versa pour les grands polygones où le nombre de vecteurs de déplacement sera plus important (voir Figure 6-4).

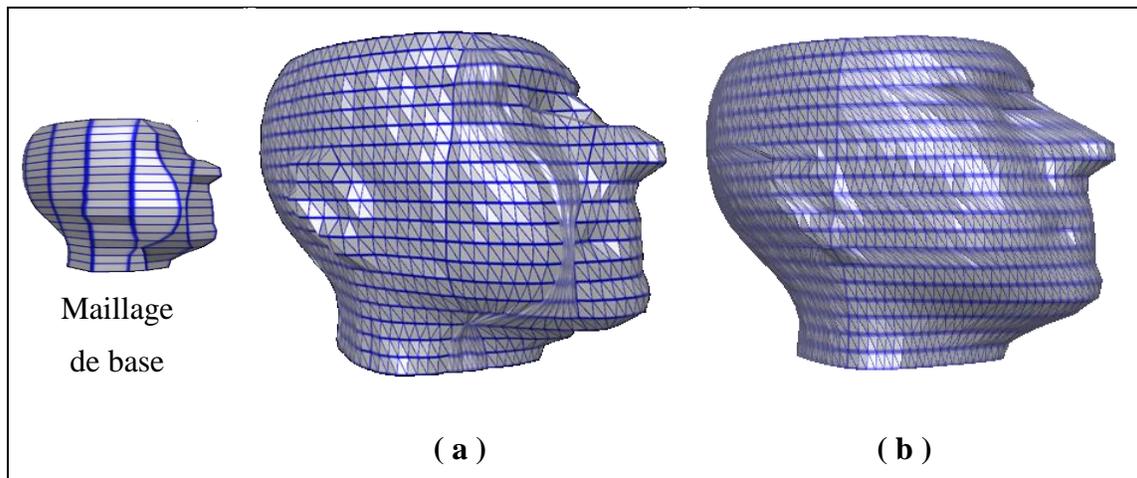


Figure 6-4 : homogénéité au niveau de détails extraits de chaque polygone.

Note : À partir du maillage de base, L'extraction des vecteurs de déplacement avec la méthode d'origine, nous donne de différents niveaux de détails dans le maillage final (a), par contre avec la deuxième version, nous aurons le même niveau (b).

Dans la 2^{ème} étape, nous comparons la performance de la nouvelle méthode d'extraction avec la méthode origine.

Avec la méthode d'extraction origine, les arêtes du maillage de base doivent être discrétisées, puis à partir de ces arêtes, nous calculons les vecteurs de déplacement pour extraire le contour externe. Pour certaines arêtes, le contour extrait est différent du contour réel. Ce problème est dû essentiellement à la géométrie de certaines zones cachées que nous ne pouvons pas détecter avec la première méthode.

Par contre avec la nouvelle méthode d'extraction, nous avons dépassé ce problème en discrétisant le contour lui-même, les vecteurs de déplacement suivent le contour externe pour balayer tous les détails existant au niveau de l'arête en cours de traitement de l'image 2D (voir Figure 6-5).

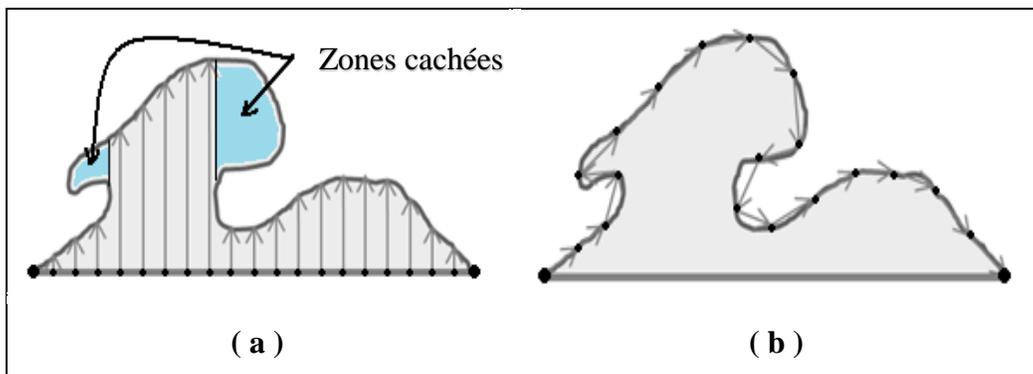


Figure 6-5 : Extraction du contour avec les vecteurs de déplacement.

Note : Avec la méthode origine, nous avons des zones cachées que ne pouvons pas détecter (a). Avec la nouvelle version, nous pouvons dépasser ce problème avec la discrétisation du contour lui-même (b).

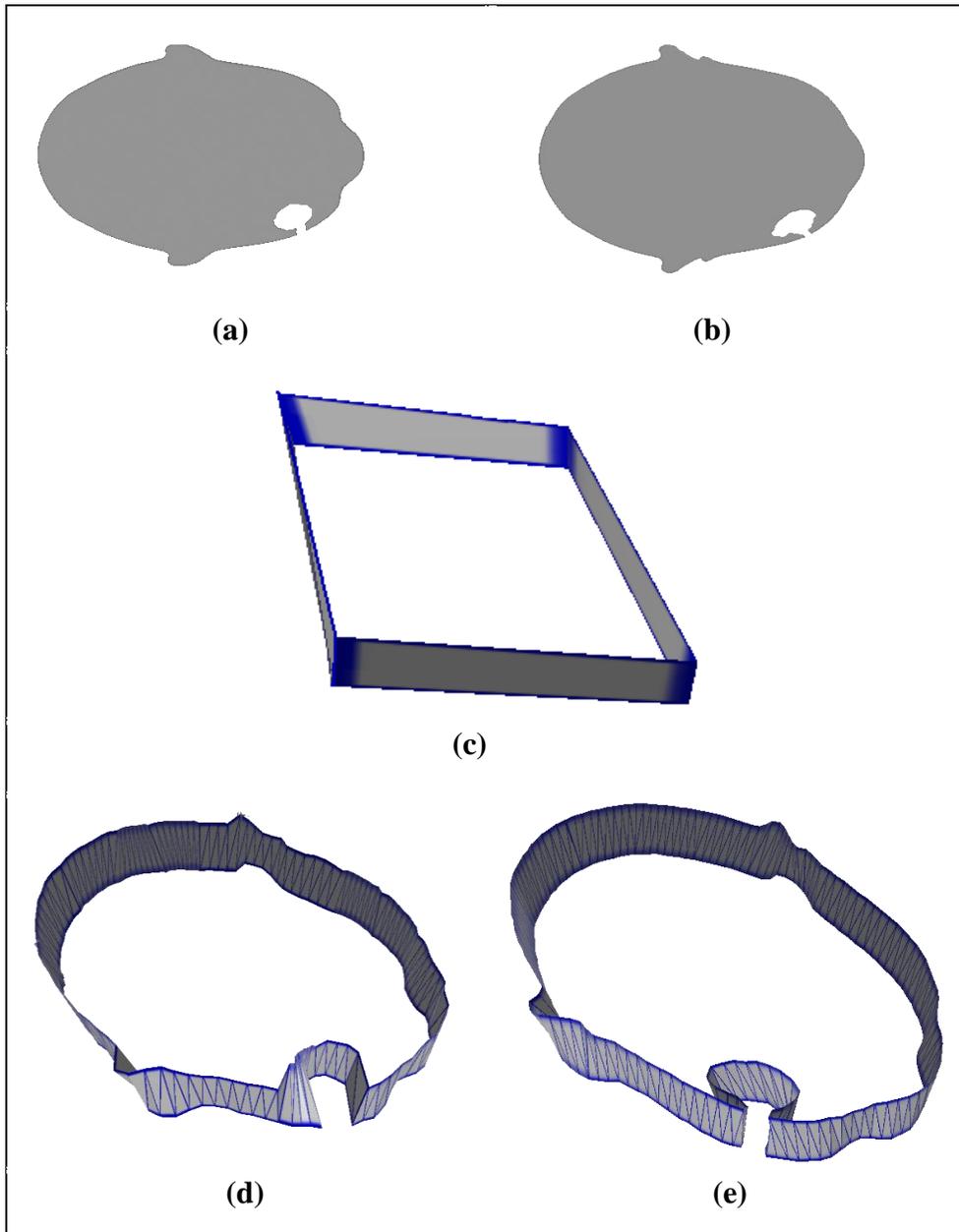


Figure 6-6 : comparaison entre les deux méthodes d'extraction.

Note : Images (a) et (b) représentent les images médicales utilisées pour construire le maillage 3D, image (c) représente le maillage de base. Après l'extraction et l'intégration des vecteurs de déplacement sur le maillage de base, nous avons obtenu l'image (d) avec la méthode origine et l'image (e) avec la nouvelle méthode.

Le nouvel algorithme d'extraction des vecteurs de déplacement exige plus de temps de calcul, mais ces calculs ne se font qu'une seule fois lors de la reconstruction du maillage 3D (calcul hors ligne).

Dans la visualisation 3D de ce maillage, nous aurons le même temps de calculs. Ces calculs sont basés sur le même maillage de base et aussi sur une texture (qui contient les vecteurs de déplacements) ayant la même dimension.

Au niveau de stockage, avec la deuxième version d'extraction, nous ne pouvons pas déterminer la taille exacte des détails à stocker, car la taille de ces détails dépend de la longueur du contour qui se trouve au niveau de chaque arête du maillage de base de l'image 2D. Mais en général, nous aurons, à peu près, la même quantité d'informations à stocker ; Pour un petit polygone du maillage de base, généralement la longueur du contour est également faible, de sorte que la quantité d'informations à mémoriser sera inférieure à celles de l'algorithme d'origine, par contre pour les grands polygones du maillage de base, il y aura le contraire : en suivant le contour de chaque arête du maillage de base, on peut certes avoir un plus grand nombre de points mais uniquement là où c'est nécessaire i.e. dans les cas complexes mais dans le cas général d'une géométrie simple au niveau de l'arête, on a moins de points au niveau des contours de l'arête à cause du nombre de vecteurs fixe dans la première méthode.

Dans le Tableau 6-2, nous faisons une comparaison entre la méthode de stockage classique du maillage haute résolution avec notre méthode qui se base sur un maillage basse résolution et sur les vecteurs de déplacement.

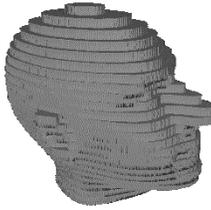
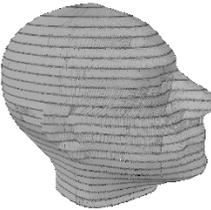
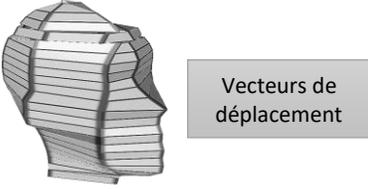
	Volume des données médicales	
	640*640*26	512*512*38
Stockage classique du maillage haute résolution obtenu par le Marching cubes	 <p>5 851 776 Octets</p>	 <p>2 480 892 Octets</p>
Stockage classique du maillage haute résolution obtenu par notre méthode	 <p>1 223 724 Octets</p>	 <p>1 393 392 Octets</p>
Stockage du maillage de base et les vecteurs de déplacement	 <p>109 468 Octets</p>	 <p>126 148 Octets</p>

Tableau 6-2 : Comparaison du stockage entre la méthode classique et la méthode qui se base sur les vecteurs de déplacement.

6.4. Conclusion

Dans ce chapitre, nous avons proposé une amélioration de l'algorithme développé dans le chapitre précédant et qui permet la reconstruction d'un maillage 3D pour les images médicales en se basant sur des vecteurs de déplacement.

L'utilisation des vecteurs de déplacement pour ajouter les petits reliefs sur un maillage de base, a prouvé son intérêt au niveau de la réduction de la quantité des

informations à stocker du maillage final 3D, et aussi au niveau de la génération automatique de la carte de déplacement qui permet d'accélérer les calculs d'affichage à l'aide de GPU.

Dans ce chapitre, nous avons modifié la méthode utilisée pour extraire les vecteurs de déplacement, cette modification a pour but d'éliminer les erreurs d'extraction, surtout avec certain type de reliefs qui ne peuvent pas être détectés avec la méthode d'origine.

Dans notre nouvelle version de la méthode d'extraction, nous avons extrait le contour de l'anatomie à reconstruire existant entre les bornes de chaque arête du maillage de base de l'image médicale. Puis nous avons généré les vecteurs de déplacement en discrétisant le contour selon le niveau de détails qu'on veut enregistrer.

Cette nouvelle version d'extraction a prouvé son intérêt en deux points principaux :

- a) La génération des vecteurs de déplacement à partir de la discrétisation du contour lui-même, permet d'augmenter la précision de ces vecteurs pour reconstruire la forme réelle de l'anatomie.
- b) La discrétisation du contour de l'anatomie avec un pas fixe, permet d'intégrer le même niveau de détails dans tous les polygones du maillage de base.

7. Conclusion générale

La visualisation médicale 3D interactive est une technique efficace pour obtenir une nouvelle vision et interagir avec les images médicales. Cette technique permet d'activer de nombreuses nouvelles techniques médicales importantes tels que la simulation chirurgicale, la chirurgie guidée par image et l'endoscopie virtuelle.

Le grand obstacle pour profiter de la visualisation médicale 3D a longtemps été celui du matériel utilisé en raison de très grandes exigences de calcul de l'ensemble de données 3D. Afin d'améliorer l'interactivité de la visualisation médicale 3D, des solutions « matériel » ont été développées. Elles consistent en des stations de travail personnalisées - et coûteuses- associées avec un matériel dédié qui permet l'interactivité en temps réel de la visualisation 3D. Ces techniques ont malheureusement un très grand coût et sont extrêmement rigides par rapport aux logiciels de l'ordinateur standard.

Dans ce contexte, nous avons présenté dans cette thèse des solutions qui permettent d'intégrer la visualisation médicale 3D dans les ordinateurs domestiques en profitant de l'évolution énorme de GPU de la carte graphique.

En précisant que toutes nos solutions respectent le plus haut degré de fiabilité médicale : Notre principe de base étant de n'afficher dans les rendus que des points issus des coupes fournies par des appareils d'imagerie médicale; nos procédés d'optimisation jouent sur le choix des points à afficher en termes de flux des données qui transitent entre CPU et GPU et sur l'exploitation parallèles de nouvelles pipelines des GPU en cherchant toujours les techniques qui offrent une bonne qualité de rendu.

L'évolution de GPU est due aux besoins des jeux vidéo : avoir une scène 3D la plus proche possible du réel avec une interactivité en temps réel. Pour satisfaire ces besoins, une nouvelle unité a été ajoutée au pipeline graphique de GPU « tessellation », qui permet de raffiner les maillages envoyés vers les GPU afin d'ajouter plus de détails avec un temps de rendu plus rapide par rapport aux autres méthodes classiques.

Dans cette thèse, nous avons développé une méthode qui permet l'exploitation de cette nouvelle architecture de GPU (en particulier l'unité de tessellation).

Parmi les techniques les plus importantes qui peuvent profiter de la tessellation matériel, nous avons le déplacement mapping. Cette technique ayant comme entrée pour les GPU : un maillage de base et une carte de déplacement. Le rôle de GPU est d'intégrer les petits reliefs enregistrés sur la carte de déplacement dans le maillage de base. Donc le défi de notre travail a été la création de ce maillage de base avec sa carte de déplacement ayant effet sur des images (coupes) médicales 2D.

A partir des algorithmes développés dans ce travail, nous sommes arrivés à remplir cette mission.

Nous avons créé le maillage de base à l'aide de l'application de Marching cubes adaptative sur un volume des données médicale. La taille des cubes utilisés change selon la complexité de la surface extraite ; ces paramètres peuvent être établis selon le domaine d'application médical une fois pour toute. Pour rendre le maillage obtenu par ce traitement utilisable par le déplacement mapping, nous avons modifié sa structure pour éliminer certains arêtes et points.

Nous avons, aussi créé la carte de déplacement automatiquement à l'aide des vecteurs de déplacement. Ces vecteurs sont calculés par l'extraction des contours de l'objet à reconstruire à partir des images médicales. A la fin de ce traitement, nous sommes arrivés à déterminer pour chaque polygone de maillage de base ses vecteurs de déplacement. Ces vecteurs de déplacement permettent d'intégrer les petits reliefs perdus pendant la construction du maillage de base.

L'implémentation de ces algorithmes dans une application nous a permis d'évaluer l'objectif de notre travail : prouver l'intérêt de l'utilisation de la carte graphique moderne afin de réduire de manière efficace, le temps d'exécution et d'augmenter la qualité de la visualisation médicale 3D sur les ordinateurs domestiques.

Dans ce travail, nous sommes arrivés aussi à obtenir une optimisation très importante dans le stockage final du maillage haute résolution. Cette optimisation est due à l'utilisation des vecteurs de déplacement dans la représentation des petits reliefs du maillage de base.

Les limites de la mise en œuvre de notre application sont directement liées à la limitation de la mise en œuvre de l'algorithme de Marching cubes, en particulier dans les cas d'ambiguïté que nous pouvons avoir dans la reconstruction partant du maillage de base. Ces cas d'ambiguïté est dû essentiellement au manque d'informations dans le volume des images médicales. Actuellement, ce problème peut être dépassé grâce à la quantité énorme des informations provenant des scanners TDI et IRM.

Cependant, nous envisageons dans la suite de ce travail d'exploiter les résultats obtenus dans la création d'une application qui incorpore toutes les étapes nécessaires pour la visualisation 3D des images médicales (recalage, segmentation, reconstruction 3D...). Cette application qui peut être installée sur les ordinateurs personnels équipés d'une carte graphique récente, et aussi d'une interface simple et des outils qui facilitent la tâche de l'utilisation surtout dans l'étape de segmentation des images.

Nous souhaitons aussi dans la suite de ce travail de passer à la visualisation 3D dynamique, c'est une visualisation qui demande plus de temps au niveau de calcul. Cette vision est alimentée principalement par les résultats obtenus dans cette thèse.

Bibliographies

- Alben, J., 2013. *NVIDIA Brings Kepler, World's Most Advanced Graphics Architecture, to Mobile Devices.* [En ligne] Available at: <https://blogs.nvidia.com/blog/2013/07/24/kepler-to-mobile/> [Accès le lundi mai 2016].
- ANDREWS, J. & BAKER, N., 2006. *Xbox 360 System Architecture.* *IEEE Micro*, 26(2), p. 25–37.
- Badano, A. et al., 2015. *Consistency and standardization of color in medical imaging: a consensus report.* *Journal of digital imaging*, 28(1), pp. 41-52.
- Baumgart., B. G., 1975. *A polyhedron representation for computer vision.* In *Proc. of the national computer conference and exposition, AFIPS (75)*, p. 589–596.
- Blinn, J. F., 1978. *Simulation of Wrinkled Surfaces.* *Computer Graphics (Proceedings of SIGGRAPH)*, 12(3), p. 286–292.
- Bloomenthal, J. & Wyvill, B., 1997. *Introduction to Implicit Surfaces.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..
- Boiselle, P. M., Reynolds, K. F. & Ernst., A., 2002. *Multipanar and three-dimensional imaging of the central airways with multidetector CT.* *AJR Am. J. Roentgenol.*, 179(2), p. 301–308.
- Boskamp, T. et al., 2004. *New Vessel Analysis Tool for Morphometric Quantification and Visualization of Vessels in CT and MR Imaging Data Sets.* *RadioGraphics*, 24(1), p. 287–297.
- Botsch, M., Kobbelt, L., M. Pauly, P. A. & Levy., B., 2010. *Polygon Mesh Processing.* CRC Press.
- Botsch, M., Steinberg, S., Bischoff, S. & Kobbelt, L., 2002. *OpenMesh: A generic and efficient polygon mesh data structure.* In *Proc. of OpenSG Symposium.*
- BOYD, C., 2008. *The DirectX 11 compute shader.* *ACM SIGGRAPH*, Volume 25.
- Canny, J., 1986. *A computational approach to edge detection.* *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), p. 679–698.
- Chernyaev, E. V., 1995. *Marching cubes 33: Construction of topologically correct isosurfaces,* Moscow: Institute for High Energy Physics.
- Congote, J. et al., 2009. *Extending Marching Cubes with Adaptive Methods to obtain more accurate iso-surfaces.* In *International Conference on Computer Vision, Imaging and Computer Graphics*, pp. 35-44.
- Cook, R., 1984. *Shade Trees.* *Computer Graphics (Proceedings of SIGGRAPH)*, 18(3), p. 223–231.

- Digabel, H. & Lantuejoul, C., 1978. *Iterative Algorithms. In Proc. of 2nd European Symposium on Quantitative Analysis of Microstructures in Material Science, 1(1), p. 85–99.*
- Doraiswamy, H. & Natarajan, V., 2013. *Computing Reeb graphs as a union of contour trees. IEEE transactions on visualization and computer graphics, 19(2), pp. 249-262.*
- Dürst., M., 1988. *Letters: Additional References to Marching Cubes. Computers & Graphics, 22(2), pp. 72-73.*
- Fiebich, M. et al., 1999. *Automatic bone segmentation technique for CT angiographic studies.. Journal of Computer Assisted Tomography, 23(1), p. 155–161.*
- Fuchs, H., Kedmen, Z. & Uselton, S., 1977. *Optimal Surface Resconstruction from Planar Contours.. Communications of the ACM, 20(10), p. 693–702.*
- Gibson, S., 1998. *Using distance maps for accurate surface representation in sampled volumes. in Proc. of Symposium on Volume Visualization, p. 23–30.*
- Ginsburg, D., Purnomo, B., Shreiner, D. & Munshi, A., 2014. *OpenGL ES 3.0 programming guide. s.l.:Addison-Wesley Professional..*
- GREMSE, F., STÄRK, M. & EHLING, J., 2016. *Imalytics Preclinical: interactive analysis of biomedical volume data.. Theranostics, Volume 6 (3), p. 328.*
- Guide, D., 2013. *CUDA C PROGRAMMING GUIDE. s.l.:NVIDIA.*
- H. Young, R. B. U. C. et al., 1999. *Measurement of clinical and subclinical tumour response using [18F]-fluorodeoxyglucose and positron emission tomography: review and 1999 EORTC recommendations.. European Journal of Cancer, 35(15), p. 1773–1782.*
- Hahn., H. K., 2005. *Morphological Volumetry:Theory, Concepts, and Application to Quantitative Medical Imaging., s.l.: PhD thesis, Department of Mathematics and Computer Science, University of Bremen..*
- Hahn, H. K. & Peitgen., H. O., 2003. *IWT-Interactive Watershed Transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional grayscale images.. In Proc. of SPIE Medical Imaging, 5032(1), p. 643–653.*
- Ho, C. et al., 2005. *Cubical marching squares: Adaptive feature preserving surface extraction from volume data. In Computer graphics forum , 24(3), pp. 537-545.*
- Hong, W. et al., 2006. *Conformal virtual colon flattening.. In Proceedings of the ACM Symposium on Solid and Physical Modeling, p. 85–93.*
- J. Sobocinski, et al., 2013. *The Benefits of EVAR Planning Using a 3D Workstation. European Journal of Vascular and Endovascular Surgery, 46(4), pp. 418-423.*
- J.F. Barrett & N. Keat, 2004. *Artifacts in ct: Recognition and avoidance.. Radiographics, 24(6), p. 1679–1691.*
- J.Wood, Z., Hoppe, H., Desbrun, M. & Schröder, P., 2004. *Removing excess topology from isosurfaces.. ACM Transactions on Graphics, 23(2), p. 190–208.*

- Jianu, R., Demiralp, C. & Laidlaw, D., 2009. Exploring 3D DTI fiber tracts with linked 2D representations.. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), p. 1449–1456.
- Johnson, C. & Hansen, C., 2004. *The Visualization Handbook*. Elsevier.
- Ju, T., Losasso, F., Schaefer, S. & Warren, J., 2002. Dual contouring of hermite data. in *Proc. of ACM SIGGRAPH*, p. 339–346.
- Kalender, W. A., R. Hebel & J. Ebersberger, 1987. Reduction of CT artifacts caused by metallic implants.. *Radiology*, Volume 164, pp. 576-577.
- Keppel, E., 1975. Approximating Complex Surfaces by Triangulation of Contour Lines.. *IBM Journal of Research and Development*, 19(1), p. 2–11.
- Kersten-Oertel, M., Jannin, P. & Collins, D. L., 2013. The state of the art of visualization in mixed reality image guided surgery.. *Computerized Medical Imaging and Graphics*, 37(2), pp. 98-112.
- Kniss, J., Premoze, S., Hansen, C. & Ebert, D., 2002. Interactive Translucent Volume Rendering and Procedural Modeling. In *Proc. of IEEE Visualization*, p. 109–116.
- Kobbelt, L., Botsch, M., Schwanecke, U. & Seidel, H.-P., 2001. Feature sensitive surface extraction from volume data. in *Proc. of ACM SIGGRAPH* , p. 57–66.
- Laligant, O. & Truchetet, F., 2010. A nonlinear derivative scheme applied to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2), p. 242–257.
- Lee, S. et al., 2003. Preoperative simulation of vascular anatomy by three-dimensional computed tomography imaging in laparoscopic gastric cancer surgery. *Journal of the American College of Surgeons*, 197(6), p. 927–936.
- Lindemann, F. & Ropinski, T., 2011. About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering.. *IEEE Transactions on Visualization and Computer Graphics*, 17(2), p. 1922–1931.
- Lopes, A. & Brodlie, K., 2003. Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Transactions on Visualization & Computer Graphics*, 9(1), p. 16–29.
- Lorensen, W. E. & Cline, H. E., 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. of ACM SIGGRAPH*, p. 163–169.
- M.Yazdi & L. Beaulieu., 2008. Artifacts in Spiral X-ray CT Scanners: Problems and Solutions.. *International Journal of Biological and Life Sciences*, 4(3), pp. 135-139.
- Mairal, J. et al., 2008. Discriminative sparse image models for class-specific edge detection and image interpretation. *Computer Vision–ECCV* , pp. 43-56.
- Meyers, D., Skinner, S. & Sloan, a. K., 1992. Surfaces from contours. *ACM Transactions on Graphics*, 11(3), p. 228–258.
- Miyawaki, Y. et al., 2008. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders.. *Neuron*, 60(5), pp. 915-929.

- Natarajan, B., 1994. *On generating topologically consistent isosurfaces from uniform samples.* *The Visual Computer*, 11(1), p. 52–62.
- Nielson, G., 2003. *On marching cubes.* *IEEE Transactions on Visualization & Computer Graphics*, 9(3), p. 283–297.
- Nielson, G. & Hamann, B., 1991. *The asymptotic decider: resolving the ambiguity in marching cubes.* in *Proc. of IEEE Visualization*, p. 83–91.
- Nießner, M. et al., 2016. *Real-Time Rendering Techniques with Hardware Tessellation.* In *Computer Graphics Forum* , 35(1), pp. 113-137.
- QUALCOMM, 2013. *Qualcomm Technologies Announces Next Generation Qualcomm Snapdragon 805 “Ultra HD” Processor.* [En ligne] Available at: <https://www.qualcomm.com/news/releases/2013/11/20/qualcomm-technologies-announces-next-generation-qualcomm-snapdragon-805> [Accès le jeudi mai 2016].
- Sabo-Napadensky, I. & O. Amir., 2005. *Reduction of scattering artifact in multislice CT.* In *Proc. of SPIE Medical Imaging : Physics of Medical Imaging.. SPIE.*
- Schroeder, W., Martin, K. & Lorensen., B., 2001. *The Visualisation Toolkit.* 3 éd. s.l.:Kitware.
- Segal, M. & A. K., 2012. *The OpenGL Graphics System: A Specification (Version 4.0 (Core Profile).* s.l., s.n.
- Serra., J., 1982. *Image Analysis and Mathematical Morphology..* London: Academic Press.
- Shiozawa, M. et al., 2009. *Preoperative virtual simulation of adrenal tumors.* *Abdominal Imaging*, 34(1), p. 113–120.
- Shu, R., Zhou, C. & Kankanhalli, M. S., 1995. *Adaptive marching cubes.* *The Visual Computer*, 11(4), pp. 202-217.
- SIMÉON, A., 2017. *Titan Xp : Nvidia dégage (à nouveau) la carte graphique la plus puissante du monde.* [En ligne] Available at: <http://www.01net.com/actualites/titan-xp-nvidia-degaine-a-nouveau-la-carte-graphique-la-plus-puissante-du-monde-1137271.html> [Accès le lundi mai 2017].
- Šoltészová, V., Patel, D. & Viola, I., 2011. *Chromatic Shadows for Improved Perception..* In *Proc. of Non-Photorealistic Animation and Rendering (NPAR)*, p. 105–115.
- Sonny, C., 2007. *Three-dimensional medical image registration on modern graphics processors.,* Calgary: Master’s thesis, University of Calgary,.
- Szirmay, k. L. & Umenhoffer, T., 2008. *Displacement Mapping on the GPU - State of the Art.* *Computer Graphics Forum* , 27(6), p. 1568–1592.
- Timothy, G., 2010. *The Mathematics of Medical Imaging: A Beginner's Guide.* s.l.:Springer.

- Treichel, T., Gessat, M., Prietzel, T. & Burgert, O., 2011. *DICOM for Implantations-Overview and Application. Journal of Digital Imaging*, 25(3), p. 352–358.
- Udupa, J. K. & Herman, G. T., 2000. *3D Imaging in Medicine*, s.l.: CRC Press.
- Vincent, L. & Soille, P., 1991. *Watersheds in Digital Spaces: an Efficient Algorithm Based on Immersion Simulations.. IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 13(6), p. 583–598.
- Watzke, O. & W. A. Kalender, 2004. *A pragmatic approach to metal artefact reduction in CT: merging of metal artefact reduced images.. European Radiology, Volume 14*, pp. 849-856.
- Yi, S., Labate, D., Easley, G. & Krim, H., 2009. *A shearlet approach to edge analysis and detection. IEEE Transactions on Image Processing*, 18(5), p. 929–941.
- Young, P. G. et al., 2008. *An efficient approach to converting three-dimensional image data into highly accurate computational models.. Mathematical, Physical and Engineering Sciences*, 366(1878), pp. 3155-3173.
- Zbijewski, W. & F. J. Beekman, 2006. *Efficient Monte Carlo Based Scatter Artifact Reduction in Cone-Beam Micro-CT.. IEEE Transactions on Medical Imaging*, 25(7), pp. 817-827.
- Zhang, L. & Bao, P., 2002. *Edge detection by scale multiplication in wavelet domain. Pattern Recognition Letters*, 23(14), p. 1771–1784.
- Zhang, Z., 1993. *Le Probleme de la mise en correspondance : l'etat de l'art*, s.l.: INRIA.