

UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE



PROJET DE FIN D'ÉTUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

RECONSTRUCTION/VISUALISATION 2D/3D D'UNE SÉQUENCE D'IMAGES

binarios

Experts in medical informatics

LIEU DE STAGE : SOCIÉTÉ BINARIOS

RÉALISÉ PAR : HALIM HAMZA

SOUTENU LE 13/06/2016

ENCADRÉ PAR :

MR. KHALID ABBAD
MR. BENABBOUD YOUSSEF
MR. ELHOMMANI MOUNIR
MR. BENDAROUICH TAHAR

DEVANT LE JURY COMPOSÉ DE :

PR. KHALID ABBAD
PR. MOHAMED OUZARF
PR. KHALID ZENKOUAR
PR. AICHA MAJDA

ANNÉE UNIVERSITAIRE 2015-2016

Remerciement

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à tous les membres de ma famille (mes parents, mes frères ...), qui m'ont beaucoup aidé par leurs soutient et leurs conseils durant les études jusqu'à aujourd'hui.

Je tiens à remercier vivement le professeur, Mr El moutaouakil Abdelmajid de l'université chouaib doukkali, d'avoir me faire connaitre le monde de l'informatique médicale en 2012, lors de mon projet de fin d'étude à la licence.

Je tiens à remercier énormément mon professeur Encadrant, Mr Abbad Khalid de l'université Sidi mohammed ben abdellah, qui m'a beaucoup aidé dans ma recherche se stage et m'a permis de postuler dans cette entreprise. Son écoute et ses conseils m'ont permis de cibler mes candidatures, et de trouver ce stage qui était en totale adéquation avec mes attentes et l'objectif de ma vie.

Je tiens à remercier mon maître de stage, Mr benabbud Youssef directeur général, responsable et chef de projet au sein de la société BINARIOS, pour son accueil, le temps passe ensemble et le partage de de son expérience au quotidien. Grace aussi à sa confiance j'ai pu m'accomplir les deux premiers étapes (sur trois) du projet final. Il fut d'une aide précieuse dans les moments les plus délicats.

Et je tiens à remercier toutes les membres des équipes :

- PACS
- RIS
- HIS
- Tous les autres personnels de l'entreprise.

Et finalement, je tiens à remercier toute l'équipe de l'enseignement du département informatique.



Résumé

Afin de faciliter l'intégration professionnelle, d'approfondir mes connaissances et de mettre en pratique les acquis de ma formation.

Le présent rapport résume mon travail accompli dans le cadre du Stage de Projet de Fin d'étude, pour l'obtention du diplôme, de Master en Sciences et Techniques des Systèmes Intelligents & Réseaux.

La mission principale de ce projet est la création d'une plateforme Web pour la reconstruction de volume 3D d'une séquence d'images médicales et la visualisation du volume suivant les différentes approches 3D.

Afin de réussir l'implémentation de la solution, j'ai opté l'utilisation des langages de programmation C/C++, HTML5/javascript et la technologie JavaEE, pour la transformation d'une séquence d'images médicales en volume manipulable prêt à être utilisé pour le diagnostic. Par le biais d'un ensemble de concepts mathématiques, tout en respectant le standard de modélisation UML.

Mots clés : DICOM, infographie, UML, approche 3D, Marching cubes, Min-IP/MIP, MultiPlanaR.



Abstract

To facilitate the professional integration, to deepen my knowledge and put into practice everything i learned using my training, a training period of study is important.

This report summarizes my work in the study of the Stage Project for graduation, to obtain a Master diploma in Sciences and Techniques for Intelligent Systems and Networks.

The main mission of this project is to create a web platform for the 3D volume reconstruction of a sequence of medical images and visualization of volume according to the different 3D approaches.

To successfully implement the solution, I opted the use of C/C++, HTML5/JavaScript programming languages and JavaEE technology for transforming a sequence of medical images in volume handled ready to be used for diagnostics. Through a set of mathematical concepts, while respecting the UML standard.

Keywords: DICOM, infographics, UML, 3D approach, Marching cubes, Min-IP/MIP, MultiPlanaR.



Table des matières

Remerciement	1
Résumé	2
Abstract	3
Tables des matières	4
Liste des figures	5
Liste des tableaux	7
Introduction générale	8
Présentation de la société	9
Introduction	9
Secteur d'activité	9
Partenariat	10
Structure de la société (organigramme incomplet)	10
Références	10
Solutions	10
Imagerie médicale	11
Introduction	11
Modalités	11
Norme DICOM	16
INTRODUCTION	16
POURQUOI UN FORMAT SPECIFIQUE POUR LES IMAGES MEDICALES ?	16
LE FORMAT EST INDEPENDANT DES MACHINES ET DES PROTOCOLES DE COMMUNICATION :	17
NOTIONS DE BASES	17
INFORMATIONS CONTENUES DANS UN FICHER DICOM	18
LE FORMAT DE FICHER DICOM	20
Infographie 3D	25
INTRODUCTION	25
LE FONCTIONNEMENT DE L'INFOGRAPHIE 3D	25
LE ROLE DE L'INFOGRAPHIE 3D DANS LE TRAITEMENT MEDICAL	30
Présentation du travail	33
Présentation du besoin :	33
Etude de besoin	33
POURQUOI CHOISIR LA 3EME SOLUTION ?	34
DESCRIPTION DE LA SOLUTION	34
Les informations nécessaires à la reconstruction 3D	36
Notion de base	38
AJUSTEMENT DE CONTRASTE	38
DIFFERENTES APPROCHES DE VISUALISATIONS 3D	39
Processus de reconstruction	42
Visualisation	45
Problème de ressources	50
Présentation de la deuxième solution	51
Conclusion	55
Références	56

Liste des figures

Figure1 : organigramme de la société.....	10
Figure 2 : produits de la société	10
Figure 3 : acquisition TDM.....	14
Figure 4 : profil d'absorption	14
Figure 5 : profil d'absorption sur plusieurs angles.....	15
Figure 6 : pixel 3D (voxel)	15
Figure 7 : parties du standard DICOM.....	17
Figure 8 : organisation des fichiers DICOM	20
Figure 9 : organisation de DATA SET	21
Figure 10 : exemple d'un pixel en bigEdian	23
Figure 11 : exemple d'un pixel en littleEdian	23
Figure 12 : vertices, arrêtes et faces	25
Figure 13 : face orientation	25
Figure 14 : translation de point.....	26
Figure 15 : homothétie de centre O.....	26
Figure 16 : rotation autour de l'axe X	27
Figure 17 : les angles alpha, beta pour la rotation	27
Figure 18 : rotation autour de l'axe Y	27
Figure 19 : rotation autour de l'axe Z	27
Figure 20 : lancer rayon, rayon primaire	29
Figure 21 : lancer rayon, intersection avec un objet	29
Figure 22 : lancer rayon, réflexion sur le point de contact	30
Figure 23 : lancer rayon, réflexion sans lumière	30
Figure 24 : pixels, voxels.....	30
Figure 25 : exemple de zone d'intérêt	31
Figure 26,27,28 : comparaison entre visualisation 3D et 2D	33
Figure 29 : diagramme de cas d'utilisation	34
Figure 30 : diagramme de séquence	35
Figure 31 : intervalle des valeurs Hounsfield	37
Figure 32 : ajustement de contraste	38
Figure 33 : effet de niveau d'ajustement	38
Figure 34 : volume vs surface	39
Figure 35 : extraction de zone d'intérêt	39
Figure 36, 37 : coupe normale, coupe oblique (courbe)	40
Figure 38 : Maximum Intensity Projection	40
Figure 39 : exemple de visualisation avec MIP	40
Figure 40 : 41 configurations pour le marching cubes	41

Figure 42 : principe de reconstruction 3D.....	42
Figure 43 : représentation d'un volume en mémoire	43
Figure 44 : distance least, distance last	43
Figure 45 : décalage des indices	44
Figure 46 : format des plans MPR	45
Figure 47 : principe de reconstruction d'un plan MPR	45
Figure 48 : reconstruction d'un plan oblique	46
Figure 49 : formation d'un cube	47
Figure 50 : représentation et numérotation d'un cube	48
Figure 51 : détermination de l'index	48
Figure 52 : diagramme de séquence	51
Figure 53 : mécanisme de communication à travers les mémoire partagée	52

Liste des tableaux

Tableau 1 : différentes identifications d'un fichier DICOM	16
Tableau 2 : différentes classes de service	18
Tableau 3 : exemple de DATA ELEMENT	19
Tableau 4 : organisation des flux de données	20
Tableau 5 : DATA ELEMENT avec VR explicite égale à (OB, OW, OF, SQ, UT, UN)	22
Tableau 6 : DATA ELEMENT avec VR explicite égale au complémentaire	22
Tableau 7 : DATA ELEMENT avec VR implicite	22
Tableau 8 : quelques exemples de transfer syntax UID	36
Tableau 9 : prétest de temps de réponse de la deuxième solution	53



Introduction générale

De nos jours, grâce à l'évolution technologique dans le domaine médical, les établissements médicaux spécialisés sont amenés à traiter des images et des volumes exponentiels, par conséquent, les professionnels de la santé expriment un besoin d'une solution permettant un traitement à la fois facile et efficace. Une solution qui doit tenir compte du développement exponentiel de la technologie d'information et de communication (laptop, tablette, mobile...), et apporter une réponse à la problématique de la multitude des plateformes (Windows, MacOS, Linux et systèmes Mobiles) dans le but de permettre aux professionnels de réaliser leur travail à n'importe quel moment, et ce quel que soit l'endroit où ils se trouvent.

Ce travail consiste à réaliser une application web permettant la visualisation 2D/3D selon les différentes approches (MPR, surface Rendering, volume Rendering, MIP, Min-IP), à partir d'une séquence d'images DICOM, dans le but d'aider le médecin radiologue à faire son diagnostic sans se limiter juste aux outils installés aux cliniques, mais il peut continuer son travail en dehors, sans avoir besoin de la machine console installée avec la modalité d'acquisition.

Dans ce contexte, nous allons procéder comme suit :

Chapitre 1 : Présentation de la société

Dans ce chapitre, je présente la société au sein de laquelle j'effectuais mon stage de fin d'étude.

Chapitre 2 : Imagerie médicale

Dans ce chapitre, je présente quelques notions importantes pour la compréhension du travail, à savoir : les différentes modalités d'acquisition, la norme DICOM et les notions de base de l'infographie 3D.

Chapitre 3 : Présentation du travail

Dans ce chapitre je présente une conception détaillée des différents cas d'utilisation et les diagrammes de séquence, ainsi que la chronologie d'implémentation pour la reconstruction et la visualisation 3D.



Chapitre 1 : Présentation de la société

1.1 Introduction

La société BINARIOS, fait partie du groupe T2S, un groupe crée en 1992, leader au Maghreb dans le domaine de production des matériels médicales avec un chiffres d'affaire de 40M euros, certifiée par l'ISO.

La société BINARIOS est créée en janvier 2008, avec un capital social de 500 000,00 DHS et de chiffre d'affaire en 2014 de 13,2M DHS.

1.2 Secteur d'activité

La conception, le développement, le déploiement et la maintenance des Systèmes d'information médicale et d'imagerie médicale.

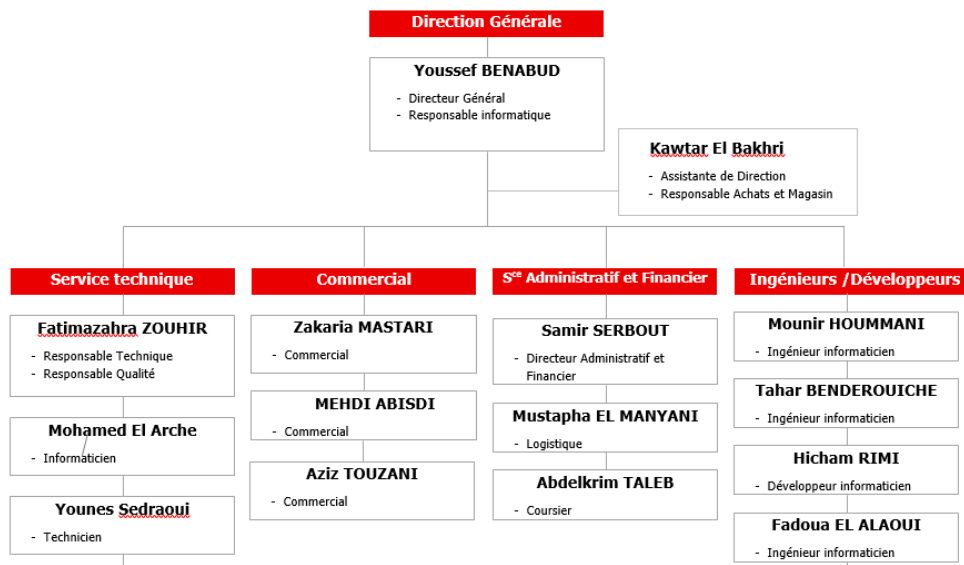
1.3 Partenariat

La société BINARIOS a pour objet principal la vente, l'installation et la maintenance de l'ensemble des produits :

- **BINARIOS** : Systèmes PACS, RIS, PACS WEB et HIS.
- **PHILIPS** : Systèmes de dictée numérique.
- **EPSON** : Robot de gravure et d'impression des CD personnalisés.
- **XEROX** : Imprimante laser couleur.
- **NUANCE** : Système de reconnaissance vocale.
- **Varay Laborix** : numériseur radiologique USB pour film RX tous formats.
- **NEC** : Ecrans médicaux de haute résolution.
- **ICR co** : system de numérisation.



1.4 Structure de la société (organigramme incomplet)



1.5 Références

Pendant les dernières années, nous avons eu la possibilité d'installer nos solutions dans certains des sites les plus prestigieux dans plusieurs pays du monde. Aujourd'hui avec une centaine d'installations en Europe et en Afrique, nous comptons parmi nos références de nombreux hôpitaux universitaires comme le Centre Hospitalier Universitaire Ibn Sina de Rabat, Casablanca, Marrakech, Oujda, Fès..., L'hôpital Militaire de Rabat, Marrakech, Agadir et Laayoun...

VisionPACS et VisionRIS sont aujourd'hui considérés comme des produits sûrs et fiables, des produits de référence dans plusieurs pays et répondent parfaitement aux besoins des structures hospitalières les plus exigeantes.

1.6 Solution

- **VISIONHIS :**
Logiciel de gestion des hôpitaux
- **VISIONPACS :**
Logiciel de gestion d'acquisition et d'archivage des images
- **TELE RADIOLOGIE – VISION WEB :**
Application web pour la visualisation des images
- **VISION RIS :**
Logiciel de gestion des cabinets de radiologie



Chapitre 2 : IMAGERIE MEDICALE

2.1 Introduction

Toute machine dédiée au secteur de l'imagerie médicale du corps humain nécessite une quelconque forme d'énergie. Dans ce cas-là, cette énergie doit être capable de pénétrer les tissus.

2.2 Modalités

Différents types d'images peuvent être obtenues en variant le type d'énergie utilisée et la technologie d'acquisition. Les différentes façons de fabriquer des images médicales sont des modalités d'imagerie. Chaque modalité a son application en médecine.

- 2.2.1 LA RADIOGRAPHIE PLANE (GRAPHIE) :

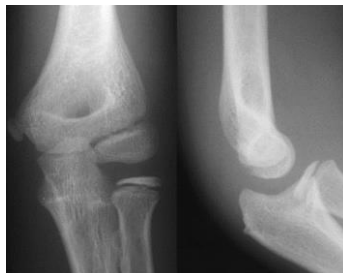
Il s'agit de la première technologie d'imagerie médicale. Elle fut découverte en 1895 par le physicien Wilhelm Roentgen, qui fit aussi les premières images de l'anatomie humaine. C'est la radiographie qui a donné naissance à la discipline appelée radiologie.



Le processus de base consiste en une émission de rayons X de courte durée (0.5 sec) de la source positionnée en face du patient et qui interagit avec celui-ci. Le détecteur permet de voir comment les rayons X se sont modifiés une fois qu'ils ont traversé le corps. Les rayons X, à la sortie de la source (du tube à rayons X) sont atténués par les milieux biologiques traversés suivant une loi exponentielle qui tient compte de l'absorption photoélectrique et de la diffusion par effet Compton. Soit I_0 le flux incident de rayons X pénétrant et suivant l'axe x dans un milieu hétérogène de coefficient d'absorption $\mu(x)$, et I le flux émergent, nous obtenons la relation suivante :

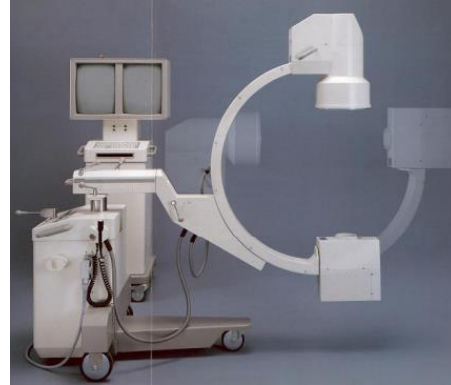
$$I = I_0 \cdot e^{-\int \mu(x) dx}$$

Dans ces principaux domaines, la radiographie est très largement utilisée pour le diagnostic de fractures osseuses, de cancer des poumons et de problèmes cardiovasculaires.



- 2.2.2 LA FLUOROSCOPIE :

La fluoroscopie est l'acquisition en continu d'images radiographiques en temps réel de radiographie. Il s'agit d'une technique qui est capable d'obtenir des images en mouvement en temps réel des structures internes du patient. Le fluoroscope est une machine équipée d'une source de rayons X et d'un écran fluorescent, laquelle est en même temps divisée en un intensificateur d'images de rayons X et une caméra CCD (caméra dotée d'un capteur CCD, dispositif à transfert de charge qui transforme les photons lumineux reçus en tensions proportionnelles au nombre d'électrons éjectés par effet photoélectrique qui seront postérieurement numérisés).



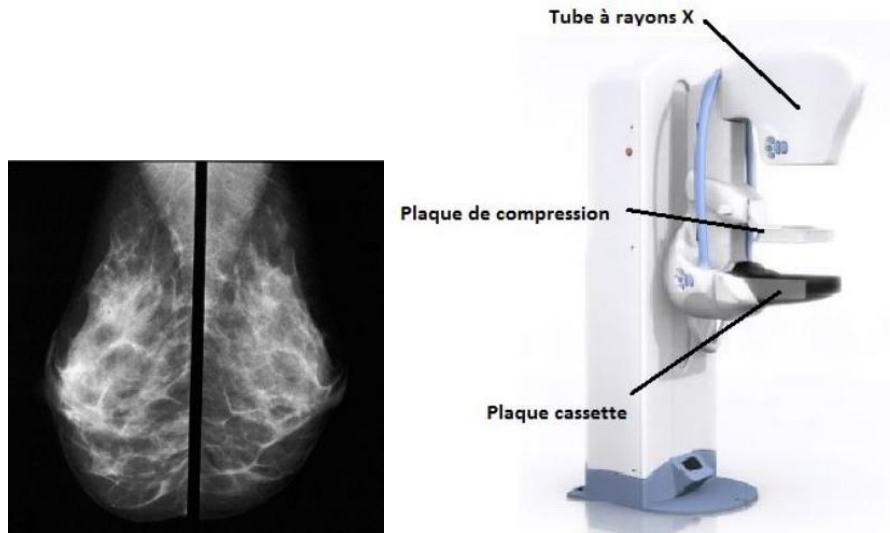
Le processus d'obtention d'images est le même que celui de la radiographie. Par contre, la fluoroscopie permet d'obtenir plusieurs radiographies simultanément et celles-ci sont rapportées sur l'écran fluorescent et enregistrées simultanément par la caméra CCD.

À titre d'exemple, l'utilisation de la fluoroscopie pour l'analyse de l'intestin grêle est mise en place à l'aide d'une radiographie conventionnelle après l'injection de produit de contraste par le biais d'un cathéter, directement dans la lumière de l'intestin grêle. La réception des rayons X transmis se fait par un écran digital, permettant d'enregistrer l'image en continu. Le point fort plus évident est l'enregistrement en continu, qui permet, en plus de la visualisation morphologique, de mettre en évidence les mouvements péristaltiques du tube digestif. Par contre, le patient reçoit une forte irradiation.



- 2.2.3 LA MAMMOGRAPHIE :

La mammographie est une radiographie de la poitrine et donc une imagerie par transmission et par projection. L'énergie des rayons X est beaucoup plus faible que celle d'autres applications (environ 0.7 mSv) et les machines modernes de mammographie sont désignées spécifiquement pour cette fin. La mammographie sert à diagnostiquer le cancer du sein asymptomatique (qui n'a pas encore produit de symptômes) et le cancer du sein symptomatique (celui qui a déjà produit symptômes).



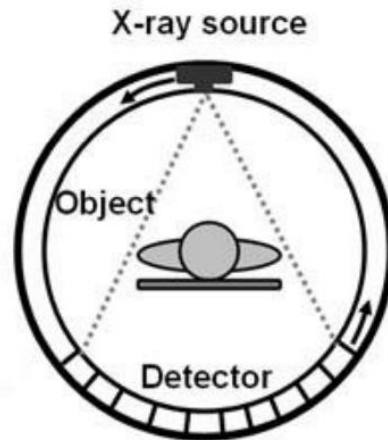
- 2.2.4 LA TOMOGRAPHIE AXIALE CALCULEE PAR ORDINATEUR :

La tomographie axiale calculée par ordinateur (TACO), aussi appelée CT-scan (CT équivaut à Computed Tomography) ou tomodensitométrie, a été disponible dès les années 70 et est la première modalité d'imagerie qui a été rendue possible grâce à l'ordinateur. Les images de tomographie axiale sont produites en passant des rayons X à travers le corps humain selon un grand nombre d'angles grâce à l'effectuation d'une rotation du tube à rayons X autour du corps. Des barrettes des détecteurs à rayons X sont situées à l'opposé de la source et collectent les données transmises. Les points ainsi collectés sont synthétisés par un ordinateur qui produit une image tomographique. Le terme de tomographie veut dire tomo (tranche) et graphie (image). Ainsi, ce processus consiste en la prise de plusieurs tranches selon des inclinaisons différentes. La tomographie axiale est une technique de transmission qui résulte en une série de tranches de tissu du patient. L'avantage est la possibilité de visualiser des tranches de tissus sans avoir la superposition des autres structures sur ou sous-jacentes. Par la suite, on pourra extraire des modèles 3D en utilisant des techniques de reconstruction.

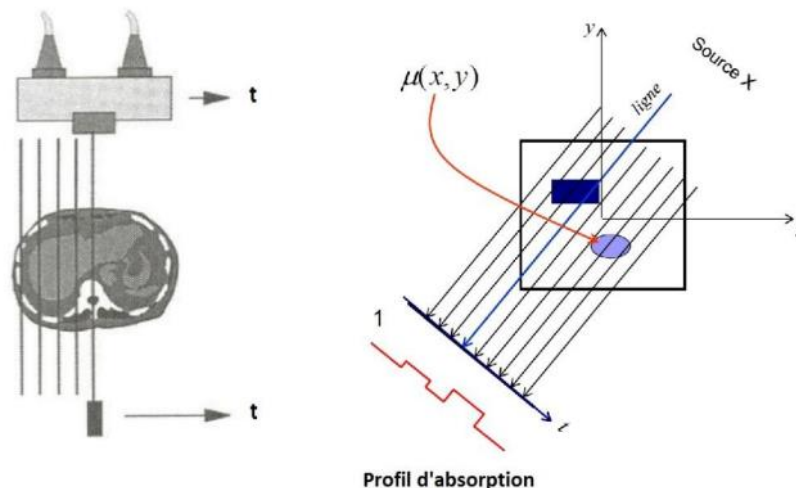


2.2.4.1 PRINCIPE D'ACQUISITION D'IMAGES :

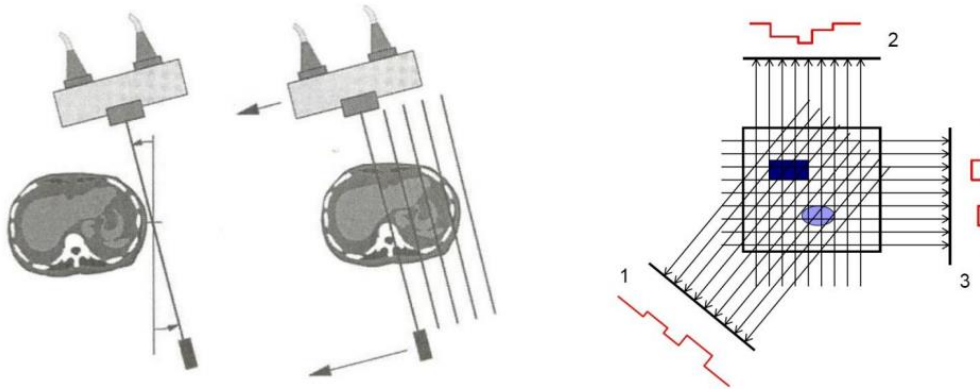
Le tomodynamomètre (CT Scan) est basé sur la mesure des différents coefficients d'absorption $\mu(x,y)$ des tissus traversés par un faisceau de rayons X (voir formule de la section radiographie) ou CA. L'absorption et l'atténuation sont deux concepts étroitement liés, car ce qui est absorbé par les tissus est dû à l'atténuation des rayons incidents. Chaque tissu a son coefficient d'absorption propre qui dépend de la densité du tissu et de l'énergie du faisceau du rayon X qui le traverse. La réalisation la plus simple d'un CT Scan nécessite donc un émetteur de rayons X, un détecteur qui lui est solidaire et un corps à étudier. Le corps sera bien sûr placé entre l'émetteur et le détecteur



En animant simultanément un mouvement de translation selon l'axe x, le faisceau de rayons X et le détecteur tournent (car ils sont solidaire). On peut connaître les projections des coefficients d'absorption en plusieurs points pour un angle donné par rapport à l'objet. C'est de cette façon que l'on obtient un profil d'absorption selon x pour un angle donné

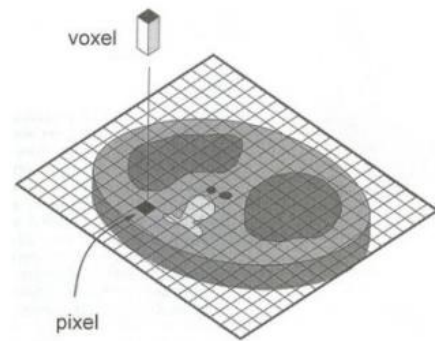


Par la suite, on fait tourner le système de quelques degrés et on recommence une série de mesures lors de la nouvelle translation



Ces opérations sont répétées sur 180 degrés.

La connaissance des profils d'absorption suivant les différents angles donne comme résultat la valeur du coefficient d'absorption à chaque point du plan. C'est un ordinateur qui se charge des calculs et qui reconstitue une image plane numérique en donnant à chaque pixel un niveau de gris correspondant au coefficient d'absorption. L'ensemble de tranches donnera différents pixels superposés, ce qui servira à l'obtention des voxels (pixels en 3D) en utilisant des traitements numériques par ordinateur des coupes 2D.



- 2.2.5 L'ÉCHOGRAPHIE :

Lorsqu'une onde ultrasonore rencontre une interface, elle est réfléchiée sans subir aucune modification de sa fréquence. L'énergie ainsi réfléchiée (écho) sera utilisée afin d'identifier, localiser et caractériser l'interface avec laquelle l'onde a frappé.

L'image échographique sera créée grâce à la captation des énergies réfléchies par les différentes interfaces traversées.

L'échographe est constitué d'une sonde, du gel, d'un système informatique, d'une console de commande, d'un système de visualisation (moniteur) et d'un système d'enregistrement des données.

Il existe plusieurs applications dans le domaine de l'échographie : échographie gynécologique et obstétricale, échographie de l'appareil locomoteur, ...



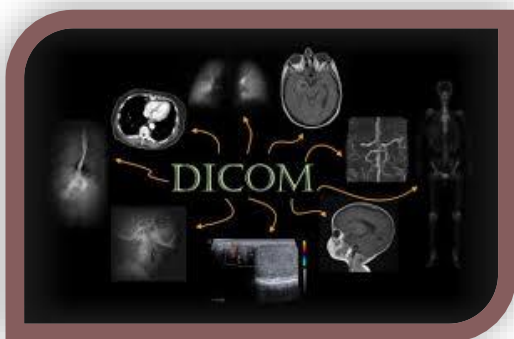
2.3 Norme DICOM

• 2.3.1 INTRODUCTION

Les croissantes évolutions des systèmes d'acquisition d'images, des systèmes d'archivage et d'information dans le cadre médical, ont produit dans les années 80 d'important besoin en connectivité et en interopérabilité des équipements médicaux. Afin d'aider à la manipulation et à la visualisation d'images, les professionnels du médical (notamment les radiologues) et les fabricants d'équipements médicaux ont développé dans un effort international commun le standard DICOM, Digital Imaging Communication in Médecine. La norme a été créée par l'ACR (American College of Radiology) en association avec la NEMA (National Electrical Manufacturers Association).

• 2.3.2 POURQUOI UN FORMAT SPECIFIQUE POUR LES IMAGES MEDICALES ?

2.3.2.1 IDENTIFICATION UNIQUE DES IMAGES PRODUITES :



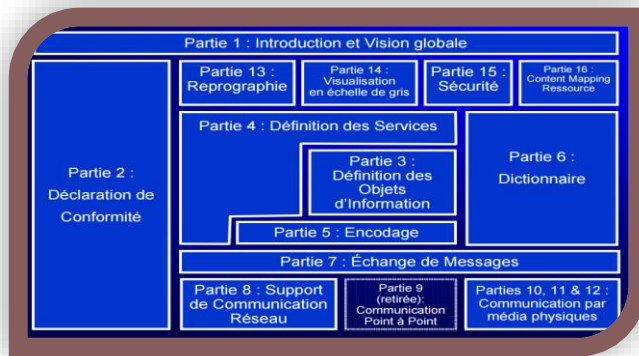
Un Service de Radiologie produit plusieurs milliers d'images chaque jour, ainsi, un scanner, travaillant au rythme de 3 patients par heure produit environ 150 images par heures, et il n'est pas possible de classer ces images dans un format courant de type JPEG ou GIF car il aurait un risque de pertes des données démographiques de l'images, (nom du patient, type d'examen, hôpital, date d'examen, type d'acquisition etc....). Le format DICOM permet de rendre unique chaque image produite et de leur associer des

informations spécifiques. Ainsi chaque image est autonome, si elle est perdue, reproduite ou renommée, il est toujours possible d'identifier formellement son origine, le patient, la date, la série d'où elle provient, les paramètres d'acquisition etc...

4 Numéros d'identifications uniques obligatoires pour chaque image DICOM :

SOP Class UID	Identifie le type de Service auquel est destiné l'image. <ul style="list-style-type: none"> • Storage Service Class • Query/Retrieve Service Class.
Study Instance UID	Identifie un examen entier, en temps et lieu.
Series Instance UID	Identifie une série d'images au sein de l'examen.
SOP Instance UID ou Image UID	Identifie l'image associée au fichier.

2.3.2.2 LA NORME PEUT ETRE FACILEMENT REVISEE ET MODIFIEE :



Actuellement la norme DICOM 3 se compose de 12 parties auxquelles s'ajoutent des suppléments. La rédaction de cette norme suit des directives strictes établies par le comité ISO.

- 2.3.3 NOTIONS DE BASES

2.3.3.1 BASE ELEMENTAIRE :

La norme DICOM est orientée objet, cela signifie que chaque objet DICOM (le plus souvent une image) contient à la fois les informations (le nom du patient, les pixels de l'image...) et les méthodes (ou fonctions) que doit subir cette information.

Exemple :

- Information : image IRM ou scanner ..., en langage DICOM : Information Object.
- Méthodes : à imprimer et/ou à sauvegarder, en DICOM un Service.

Ainsi la combinaison d'un " Information Object " (par ex une image) avec un "Service" (par exemple l'impression de cette image) est appelée : **Service/Object Pair** ou **SOP**.

Information Object + Service Class = Service /Object Pair ou SOP

Ou encore, par exemple :

Une Image + Son Impression = Un service DICOM

2.3.3.2 CLASSES DE SERVICE ACTUELLEMENT DISPONIBLES DANS LA NORME :

Classes de Service :	Type de Service :
Verification (Verification Service Class)	Utilisé pour les tests, permet de savoir si les machines "s'entendent" mutuellement, cette classe n'est pas associée à un objet DICOM, elle renvoie l'information sous la forme d'un écho. (C-ECHO)
Storage (Storage Service Class) Media Storage Service Class	Permet le transfert et la sauvegarde des images entre deux entités DICOM. (CR, CT, MR Storage Service Class) Il existe une variante : Media Storage Service Class qui spécifie les échanges entre 2 machines par l'intermédiaire d'un média (CD rom, disquettes etc....)
Query/Retrieve	Implémente des commandes types : FIND, MOVE, GET. FIND permet de demander une liste d'image, MOVE et GET permettent d'initier un transfert, qui sera en réalisé effectué via la classe "Storage Service Class"
Study Contents Notification	Utilisée pour notifier l'arrivée d'une nouvelle image ou série d'images, peut être utilisée pour initier un transfert ou vérifier si le transfert d'une série d'image est complet.
Print Management	Permet la connexion avec un reprographe, spécifie le type d'image, (Couleurs, niveaux de gris etc...)
Patient Management	Permet d'interfacer la machine au réseau hospitalier PACS ou HIS/RIS (Hospital Information Service/ Radiological Information Service) Gestion des données des patients, démographie, admission et sortie des patients
Study Management	Création, gestion de rendez-vous, suivi des examens.
Result Management	Permet la gestion des résultats des examens.

- 2.3.4 INFORMATIONS CONTENUES DANS UN FICHER DICOM

2.3.4.1 INFORMATIONS HABITUELLEMENT CONTENUES DANS UN FICHER DICOM :

Un fichier DICOM comprends les données correspondantes aux pixels de l'image, habituellement ces données sont groupées à la fin du fichier DICOM, elles sont précédées par d'autres données techniques et démographiques.



2.3.4.2 ORGANISATION DES DONNEES DANS UN FICHIER DICOM :

L'organisation de l'information contenue dans les fichiers DICOM a été inspirée par la sauvegarde des données sur des bandes magnétiques. L'information est organisée sous une forme séquentielle.

Chaque information élémentaire est constituée de 3 champs de données. Le premier champ est codé sur 8 octets, il s'agit d'une "balise" ou "tag", répertoriée dans le dictionnaire DICOM, qui indique le type d'information qui va suivre. Le deuxième champ de 8 octets indique la longueur de l'information contenue dans le 3ème champs, jusqu' à la balise suivante.

Exemple : Codage de l'identification du patient :

Si on rencontre dans un flux de données DICOM les éléments suivants en hexadécimal :

00 10 00 20 00 00 00 0A 4A 6F 68 6E 5E 48 65 6E 72 79

Ce flux se décompose ainsi :

4 octets Balise groupe	4 octets Balise élément	8 octets Longueur l du champ suivant	Longueur variable l, Information :																				
0010	0020	0000000A	4A 6F 68 6E 5E 48 65 6E 72 79 (Ici l = 10)																				
Où : 0x0010 correspond dans le dictionnaire au code hexadécimal du groupe de données codant l'information du patient.	0x0010 0010, balise du dictionnaire indiquant que l'information sera le nom du patient	Indique la longueur du champ suivant. (Ici A en hexadécimal correspond à 10 en décimal)	Ici signification = ASCII <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>4A</td><td>6F</td><td>68</td><td>6E</td><td>5E</td><td>48</td><td>65</td><td>6E</td><td>72</td><td>79</td> </tr> <tr> <td>J</td><td>o</td><td>h</td><td>n</td><td>^</td><td>H</td><td>e</td><td>n</td><td>r</td><td>y</td> </tr> </table>	4A	6F	68	6E	5E	48	65	6E	72	79	J	o	h	n	^	H	e	n	r	y
4A	6F	68	6E	5E	48	65	6E	72	79														
J	o	h	n	^	H	e	n	r	y														

Ces informations se succèdent à la suite les unes des autres dans un ordre de balises croissant, il est ainsi possible d'ignorer certaines informations contenues dans des champs d'information et de passer à la balise suivante, ce qui permet de " sauter " une balise dont on ne connaît pas la signification.

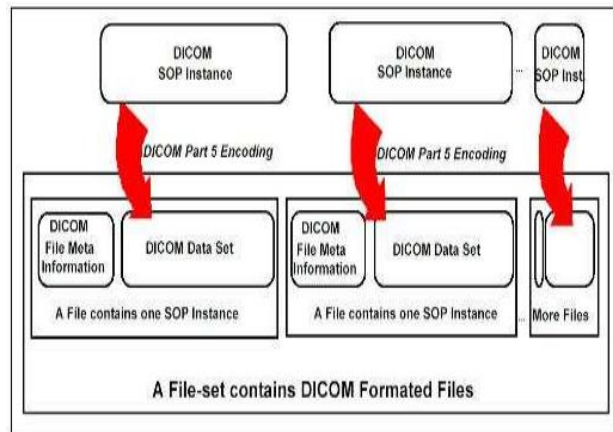
2.3.4.3 ORGANISATION DES FLUX DE DONNEES :

On trouve ainsi les données suivantes dans le flux DICOM :

L'identification de la machine	<ul style="list-style-type: none"> • Date d'examen • Type d'examen • Fabricant de la machine • Hôpital ou institution • Identification de la machine (numéro, type)
Les informations sur le patient :	<ul style="list-style-type: none"> • Nom • Identification • Date de naissance • Sexe
Les informations sur l'acquisition de l'information :	<ul style="list-style-type: none"> • Bolus • Kilovolts • Inclinaison du statif • Epaisseur de coupe • etc. variable suivant le type d'examen
Les informations en rapport avec l'examen :	<ul style="list-style-type: none"> • Série, • Orientation du patient • Plan de références • Nombres d'images dans l'acquisition • Commentaires
Les informations concernant l'image elle-même et la façon dont elle est codée :	<ul style="list-style-type: none"> • Largeur • Hauteur • Codage de l'image (niveaux de gris, couleur ...) • Compression • Représentation numérique de l'image
Les pixels :	<ul style="list-style-type: none"> • Représentation numérique de l'image elle-même.

• 2.3.5 Le format de fichier DICOM :

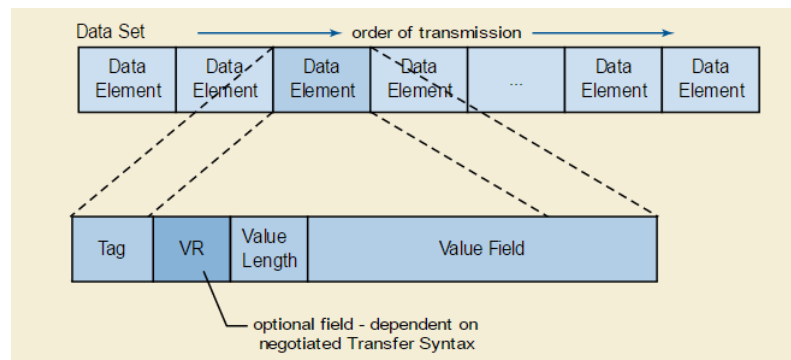
Le format de fichier DICOM fournit un moyen d'encapsuler l'ensemble des données représentée par une « Instance SOP » relative à une définition de l'objet d'une classe SOP. Les fichiers contenant une instance se succèdent et le tout constitue un ensemble de fichiers DICOM. Dans un fichier DICOM, les données sont organisées sous une forme séquentielle en commençant par une entête et suivi des données brutes de l'image.



2.3.5.1 ENTETE :

Chaque entête commence par un préambule de 128 octets généralement mis à zéro suivi de 4 octets pour y inscrire les caractères 'D'. 'I'. 'C'. 'M'.

2.3.5.2 DATA SET :



2.3.5.3 DATA TAG ELEMENT :

Est une paire ordonnée de 16 bits d'entiers non signés représentant le numéro du groupe et suivi par le numéro de l'élément. Elle indique le type d'information qui va suivre.

2.3.5.4 VALUE REPRESENTATION :

Est une chaîne de deux caractères de 2 octets associée à la valeur du Data Tag Element précédent. Elle indique la longueur de l'information contenue dans le dernier champ jusqu'au Tag suivant.

2.3.5.5 VALUE LENGTH :

Est soit un entier non signé de 16 ou 32 bits dépendant d'un VR explicite ou implicite, et contenant la longueur explicite du champ de valeur en nombre d'octet.

2.3.5.6 VALUE FIELD :

Est un champ de valeur de longueur variable. Il correspond à l'information identifié par le premier champ.



- 2.3.5.6.1 STRUCTURE D'ELEMENT DE DONNEES AVEC UN VR EXPLICITE :

Quand en utilisant la structure d'un VR explicite, l'élément de données sera construit de 4 champs consécutifs : Data Element Tag, VR, Value Length, et Value Field. Selon le VR, il y a deux constructions possibles.

Il y a donc VR égal à OB, OW, OF, SQ, UT ou UN.

Tag		VR		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string) of "OB", "OW", "OF", "SQ", "UT" or "UN"	Reserved (2 bytes) set to a value of 0000H	32-bit unsigned integer	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	'Value Length' bytes if of Explicit Length

Et il y a VR égal au complémentaire.

Tag		VR	Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string)	(16-bit unsigned integer)	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax.
2 bytes	2 bytes	2 bytes	2 bytes	'Value Length' bytes

- 2.3.5.6.2 STRUCTURE D'ELEMENT DE DONNEES AVEC UN VR IMPLICITE :

Quand en utilisant la structure d'un VR implicite, l'élément de données sera construit de 3 champs consécutifs : Data Element Tag, Value Length, et Value.

Tag		Value Length	Value
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	32-bit unsigned integer	Even number of bytes containing the Data Elements Value encoded according to the VR specified in PS3.6 and the negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	4 bytes	'Value Length' bytes or Undefined Length

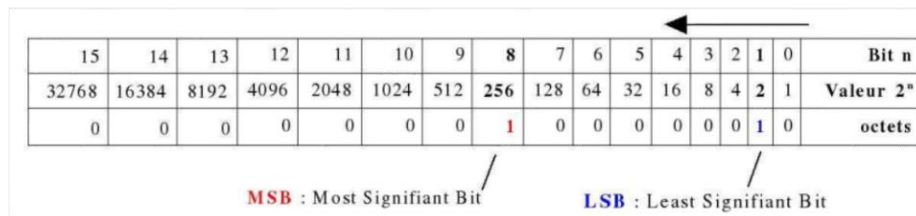


2.3.5.7 IMAGE :

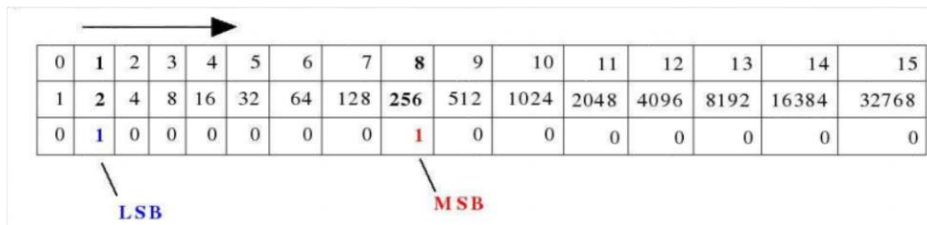
Les images sont la dernière DATA ELEMENT dans une DATA SET. Elles peuvent être compressées au format JPEG par l'intermédiaire d'un groupe d'élément « Transfer Syntax UID ». La façon dont les pixels sont sauvegardés peut énormément changer. Selon l'architecture de l'ordinateur, le problème de l'ordonnancement des bits intervient lors de la lecture (little/big endian).

Par exemple, un pixel qui a pour valeur 258 (en décimal) peut être représenté selon l'ordre « Little Endian » ou « Big Endian » en binaire.

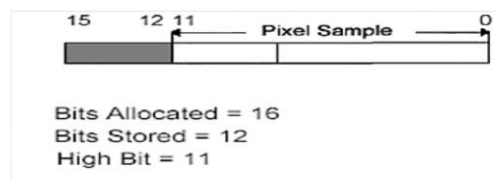
- Big Endian :



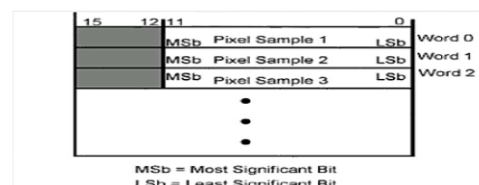
- Little Endian :

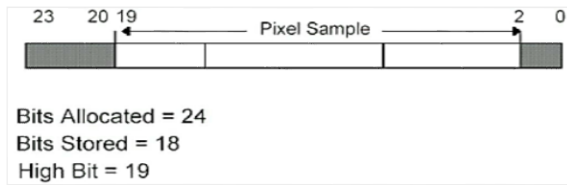


Le souci de ces modes little ou big endian se pose au niveau de l'ordonnancement des octets. Chaque pixel d'une image est codé sur n bits et ordonnancés en mot (groupement de d'octets). Dans un cas où l'information des pixels est codée sur 16 bits et contenue dans des mots de 16 bits, l'ordonnancement des mots se succèdent simplement les uns derrière les autres.



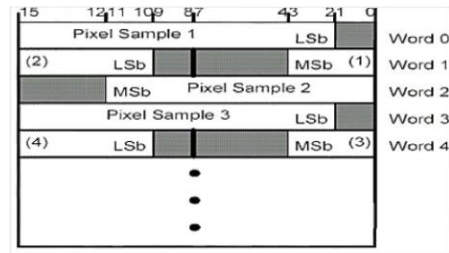
Ensuite, chaque mot se succèdent et renferme l'information d'un pixel.





Nous pouvons aussi coder l'information sur 24 bits en ayant seulement besoin de stocker 18 bits, et choisir le bit le plus significatif à la position 19. Dans cette situation l'ordonnement des mots se complique.

Ensuite, si en plus nous souhaitons ordonner ces données en mots de 16 bits, chaque échantillon de pixel sera étalé sur deux mots. (Agencement de chaque pixel sur 2 mots est relatif au VR = OW)



En général, nous retiendrons que les processeurs Intel ordonnent les octets en mode Little Endian et, les Motorola (MAC) et Sparc (SUN) en mode Big Endian.

2.4 INFOGRAPHIE 3D

- 2.4.1 INTRODUCTION

L'infographie 3D est une branche de l'informatique qui s'intéresse à la création d'images numériques de synthèse. En effet, l'infographie en général se base sur la représentation de données informatiques abstraites, sous forme d'images. L'infographie 3D en particulier est basée sur l'espace virtuel, c'est-à-dire, sur une représentation en trois dimensions des données. L'espace peut être défini par un repère (O, i, j, k) , ou bien par trois axes orthogonaux X, Y et Z, correspondant respectivement aux abscisses, ordonnées et aux cotes.

- 2.4.2 LE FONCTIONNEMENT DE L'INFOGRAPHIE 3D :

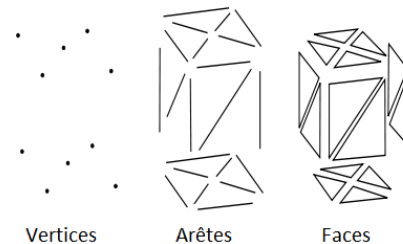
2.4.2.1 MODELISATION DANS L'ESPACE VIRTUEL 3D :

La première étape dans la synthèse d'images par infographie 3D est celle de la modélisation d'un "objet" donné.

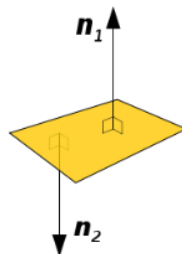
- 2.4.2.1.1 LE MAILLAGE :

Le maillage est le nom donné à tout objet modélisé dans un espace tridimensionnel. Le maillage est formé de plusieurs groupes d'éléments :

- Un vertice : C'est non seulement un point dans l'espace, mais c'est une unité à laquelle l'ordinateur associe des informations sur la position (du type $p(x; y; z)$), la couleur, etc.
- Une arête : C'est un segment qui relie deux vertices.
- Une face : Pour un logiciel d'infographie 3D, une face est un triangle composé de trois arêtes refermées et connectées par leurs sommets.



De plus, à chaque face est associée une droite normale. C'est une droite perpendiculaire au plan auquel appartient une. Et à cette droite est associé un vecteur directeur. Ici, l'intérêt que présente le vecteur directeur est de spécifier l'orientation de la face.



- 2.4.2.1.2 LA MANIPULATION D'OBJETS DANS L'ESPACE :

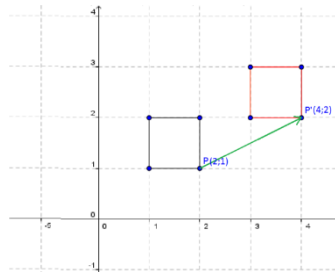
Nous avons ainsi vu que l'infographie 3D consiste à modéliser des objets virtuels. Cependant le tout est de pouvoir manipuler les maillages dans l'espace. Ceci nous mène donc à expliquer les principes mathématiques qui permettent cela.

Différents types de manipulation existent, mais nous n'allons étudier que les trois les plus utiles : La translation, l'homothétie et la rotation.

- Translation :

Une translation (vectorielle) est simplement un déplacement d'objet. Une translation de vecteur $\vec{u}(a, b, c)$, consiste à associer à un point P (x, y, z) un point P' (x', y', z') tel que :

$$X' = x + a. \quad Y' = y + b. \quad Z' = z + c$$



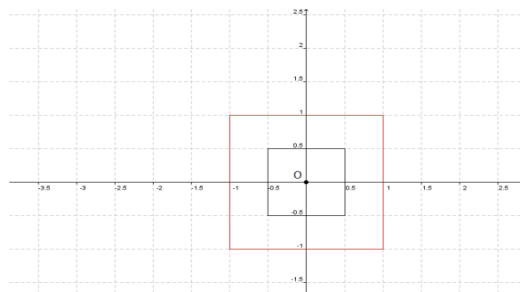
Dans l'image ci-dessus, une translation a été faite dans le plan. Le carré rouge est l'image du carré noir par la translation vectorielle $\vec{u}(2;1)$.

- Homothétie :

L'homothétie de centre O, consiste à associer à un point P (x, y, z), un point P' (x', y', z') tel que :

$$x' = x \times S_x. \quad y' = y \times S_y. \quad z' = z \times S_z$$

S_x , S_y , et S_z sont appelés des scalaires. Ce sont des réels qui « contrôlent » l'agrandissement (ou le rétrécissement) sur les différents axes du repère (respectivement l'axe des abscisses, l'axe des ordonnées et l'axe des cotes).



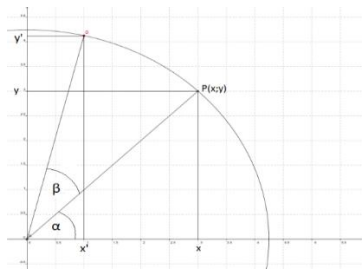
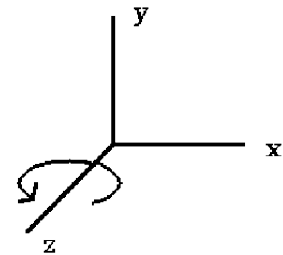
- Rotation :

La rotation consiste à faire pivoter un objet (maillage) dans l'espace, autour d'un axe donné.

- La rotation autour de l'axe des cotes :

Elle associe à un point P (x, y, z), un point P' (x', y', z'), tel que :

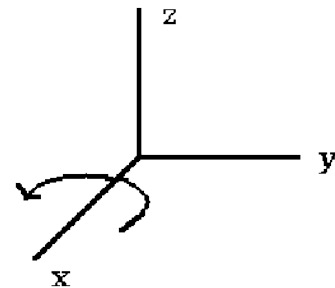
- $x' = \cos(\alpha + \beta) \times OP$. (Car $x'/OP = \cos(\alpha + \beta)$, avec
 - OP : distance de P par rapport à l'origine O
 - α : angle de rotation initial de P
 - β : angle de la rotation à effectuer
- $y' = \sin(\alpha + \beta) \times OP$
- $z' = z$



- La rotation autour de l'axe des abscisses :

Elle associe à un point P (x, y, z), un point P'(x', y', z'), tel que :

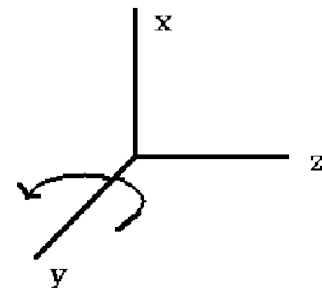
- $y' = \cos(\alpha + \beta) \times OP$, Avec
 - α : angle de rotation initial de P autour de l'axe des abscisses
 - β : angle de la rotation à effectuer autour de l'axe des abscisses
- $z' = \sin(\alpha + \beta) \times OP$
- $x' = x$



- La rotation autour de l'axe des ordonnées :

Elle associe à un point P (x, y, z), un point P'(x', y', z'), tel que :

- $z' = \cos(\alpha + \beta) \times OP$, avec
 - α : angle de rotation initial de P autour de l'axe des ordonnées
 - β : angle de la rotation à effectuer autour de l'axe des ordonnées
- $x' = \sin(\alpha + \beta) \times OP$
- $y' = y$

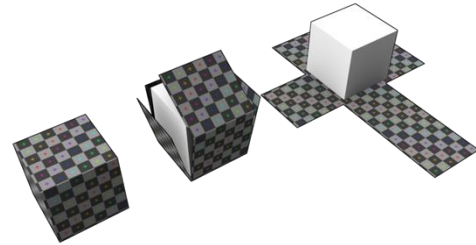


2.4.2.2 LA VISUALISATION D'UNE SCENE VIRTUELLE :

- 2.4.2.2.1 LES TEXTURES :

Les textures sont des images servant de « vêtements » pour un maillage. Mais la modélisation en 3D des objets n'est qu'une représentation de leur volume et de leur aspect géométrique.

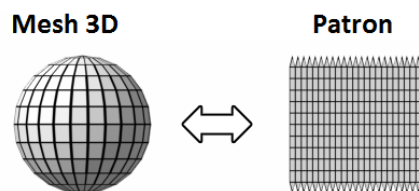
En 1974, Edwin Catmull crée une méthode appelée le Texture-mapping. Il s'agit d'appliquer une image « normale » sur la surface d'un maillage tridimensionnel. On peut imaginer par exemple une boîte (qui représenterait le maillage) et du papier cadeau (qui représenterait la texture). Le Texture-mapping consisterait à emballer la boîte avec le papier cadeau.



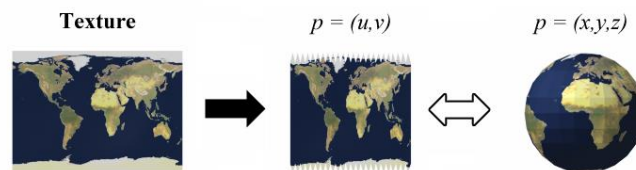
Plus exactement, le Texture-mapping se décompose en deux parties :

- Le UV Unwrapping, et
- L'application de la texture.

– UV Unwrapping : C'est un processus par lequel l'ordinateur détermine le patron d'un maillage. Effectivement, en Français, cela signifierait « déemballage en UV ». Les lettres U et V remplacent X et Y, car ces deux lettres sont déjà utilisées par l'ordinateur dans les calculs des coordonnées des vertices. Il s'agit donc de donner à chaque vertice des coordonnées (du type $P(u, v)$) dans le plan plus. Le patron est donc obtenu.



– Application de la texture : Il s'agit de placer une image sur le patron obtenu : c'est le texture-mapping. Effectivement, l'ordinateur enregistre pour chaque face, les pixels de la texture, appelés texels, qui lui appartiennent ainsi que leur position. Enfin, chaque face du maillage en 3D diffuse les texels qui lui ont été appropriés. Ainsi, la texture est appliquée sur la surface de l'objet modélisé.



- 2.4.2.2.2 LE RENDU :

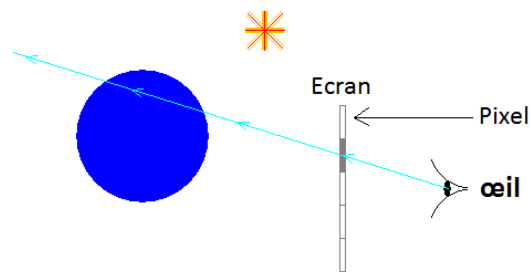
L'étape pendant laquelle un logiciel appelé moteur de rendu 3D transforme une scène virtuelle tridimensionnelle en une image normale (2D).

- Lancer du rayon :

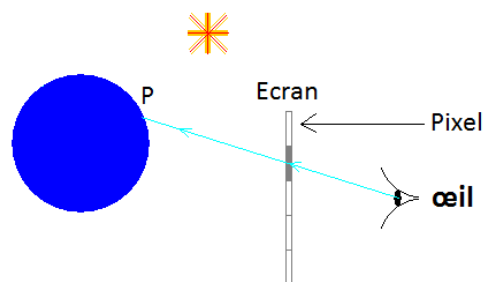
Dans la vie réelle, lorsque nous observons quelque chose, nos yeux perçoivent une image en « interprétant » les rayons lumineux qui parviennent à nos yeux. Il en est presque de même dans un logiciel de modélisation 3D. Mais on sait que la lumière est émise en une infinité de rayons et dans toutes les directions, alors il ne serait pas efficace pour un ordinateur d'effectuer des calculs pour un très grand nombre de rayons, puisque la majorité de ceux-ci ne figureront pas sur l'image finale. Il s'agit alors de suivre le trajet inverse de chaque rayon lumineux qui parvient à l'image qu'on souhaite obtenir. On parle alors de Ray-Tracing, soit « lancer de rayons ».

- Etape 1 :

Le Ray-tracing est un algorithme de rendu d'image. Effectivement pour chaque pixel de l'image que l'on souhaite obtenir, il projette un rayon, c'est un rayon primaire



L'algorithme trouve le premier point (P) d'intersection de chaque rayon primaire avec un maillage de la scène 3D.



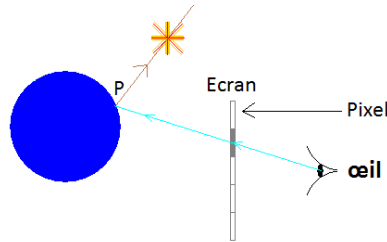
Il détermine ensuite à quel maillage, et à quelles faces du maillage ce point appartient. Ceci permet donc de ne représenter que les faces visibles.



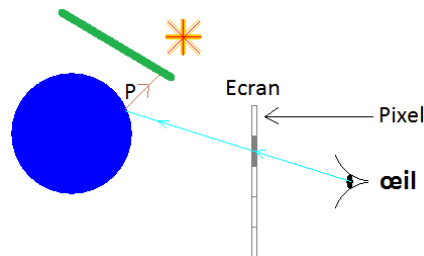
- Etape 2 :

Il faut ensuite déterminer la couleur de la face à laquelle appartient ce point.

Si aucun objet ne coupe le rayon, alors l'algorithme saisit la valeur de l'intensité de la lumière et l'affecte à la face du maillage.



Si le rayon est coupé par un autre objet, le pixel est par conséquent de couleur noire.



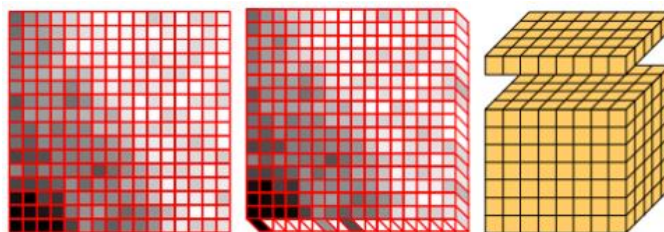
- 2.4.3 LE RÔLE DE L'INFOGRAPHIE 3D DANS LE TRAITEMENT MÉDICAL :

- 2.4.3.1 LE PASSAGE DE 2D A 3D :

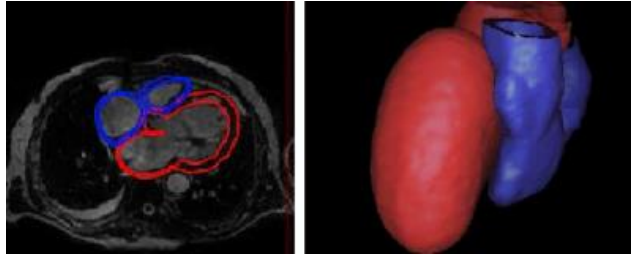
Les tomographies obtenues par les IRM ou les TDM sont de simples images (2D) monochromes.

Une image est formée de petits carrés qui émettent de la lumière, appelés pixels. Alors pour passer d'une image (2D) à un modèle 3D, une dimension est ajoutée. Les pixels (carrés) deviennent des voxels (cubes).

Les voxels sont générés pour les pixels de chaque tomographie. On obtient alors des couches de voxels pour obtenir une forme tridimensionnelle.

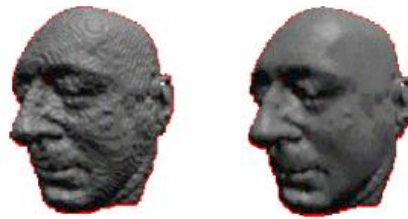


Cependant, pour recréer un modèle en 3D, il est nécessaire de déterminer la région d'intérêt à visualiser : c'est la segmentation



C'est une étape relativement simple où l'utilisateur marque la région d'intérêt. Sachant que les différents tissus (organes) sont représentés par des pixels d'intensité différente sur la tomographie.

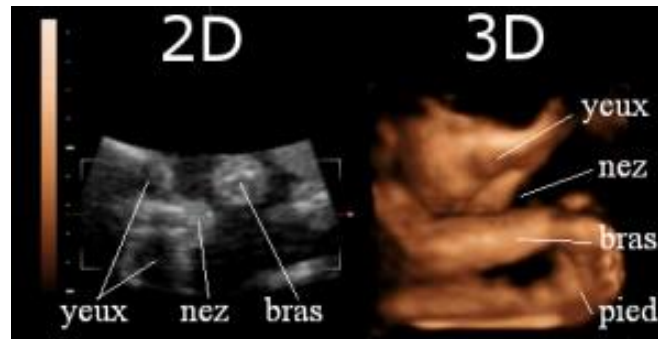
Enfin, les couches de voxels sont empilées, et une forme tridimensionnelle est obtenue.



2.4.3.2 AVANTAGES DE L'IMAGERIE 3D PAR RAPPORT A LA 2D :

- EN ECHOGRAPHIE :

La modélisation en 3D offre l'avantage d'une visibilité accrue des os du fœtus. Ceux-ci forment une structure assez complexe, notamment avec la colonne vertébrale et les côtes, ce qui peut être difficile à distinguer, et à étudier en échographie 2D.



- EN TOMOGRAPHIE :

La modélisation en 3D présente des avantages par rapport aux tomographies obtenues par IRM ou TDM, en termes de facilité de visualisation.

Dans le cas des tomographies, le médecin doit visualiser plusieurs coupes successives du corps ; le repérage des organes devient difficile, et par conséquent l'analyse des anomalies l'est aussi. Nous pouvons clairement voir que, l'imagerie 3D permet une lisibilité accrue.

Les reins par exemple, peuvent être facilement distingués dans le modèle 3D.



Chapitre 3 : Présentation du travail

3.1 Présentation du besoin :

La radiologie médicale, désigne un ensemble des modalités utilisant les rayonnements (radiation désigne le processus d'émission ou de transmission d'énergie) qui ont pour objectif de produire des images du corps humain (imagerie médicale).

Un médecin radiologue est un médecin spécialiste des techniques de l'énergie médicale (cliches radio, échographies, scanners et IRM). Il repère notamment les fractures, tumeurs ou infections sur toutes les parties du corps. Il surveille la croissance des fœtus chez les femme enceinte....

D'où la nécessité d'un outil permettant la visualisation des images et propose la puissance des traitements d'images devant ces médecins pour les aidés à faire leurs travaux dans les meilleures conditions.

Alors le besoin se focalise autour d'un outil puissant, simple à l'utiliser (pas d'installation complexe, moins de configurations ...), et vue le développement de la technologie (laptop, mobile, tablette ...), ce besoin s'est développe vers un outil qui marche pour ces technologies et non pas juste pour les stations de haute performance (work Station).

3.2 Etude de besoin

Cette étude est réalisée par une équipe technique de quatre personne :

- Benaboud Youssef (directeur de la société BINARIOS)
- El hommani Mounir
- Bendarouich Tahar
- Halim Hamza

Après une recherche profonde, pour engendrer le besoin, on s'est trouvé face au problème de multiplateforme :

- Windows (bureau, mobile).
- Mac os et ios
- Linux
- Android
- ...



Les solutions proposées pour résoudre ce problème sont :

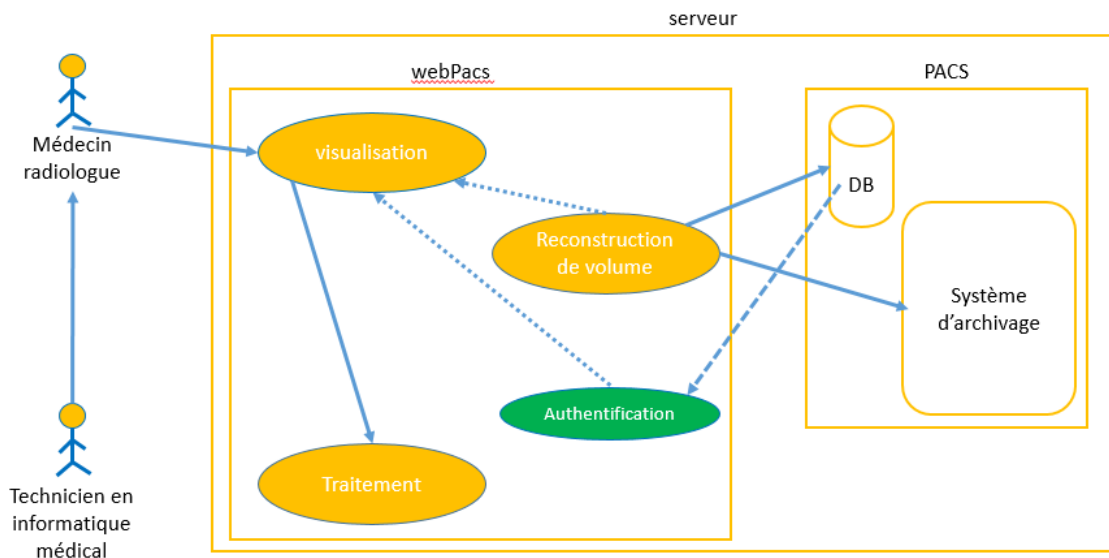
- La première solution est de réaliser une application par plateforme. Une Telle solution consomme de l'énergie, nécessité de la main d'œuvre et surtout beaucoup de temps ainsi que le grand problème de maintenance et d'amélioration par la suite.
- La deuxième solution est l'utilisation de la plateforme XAMARIN proposée par Microsoft qui permet de générer trois applications pour les trois plateforme mobile en utilisant le même code source. Mais toujours on a un manque pour les autres plateformes.
- La troisième solution (solution adoptée) est celle d'une application web qui ne nécessite qu'un navigateur.

- 3.2.1 POURQUOI CHOISIR LA 3EME SOLUTION ?

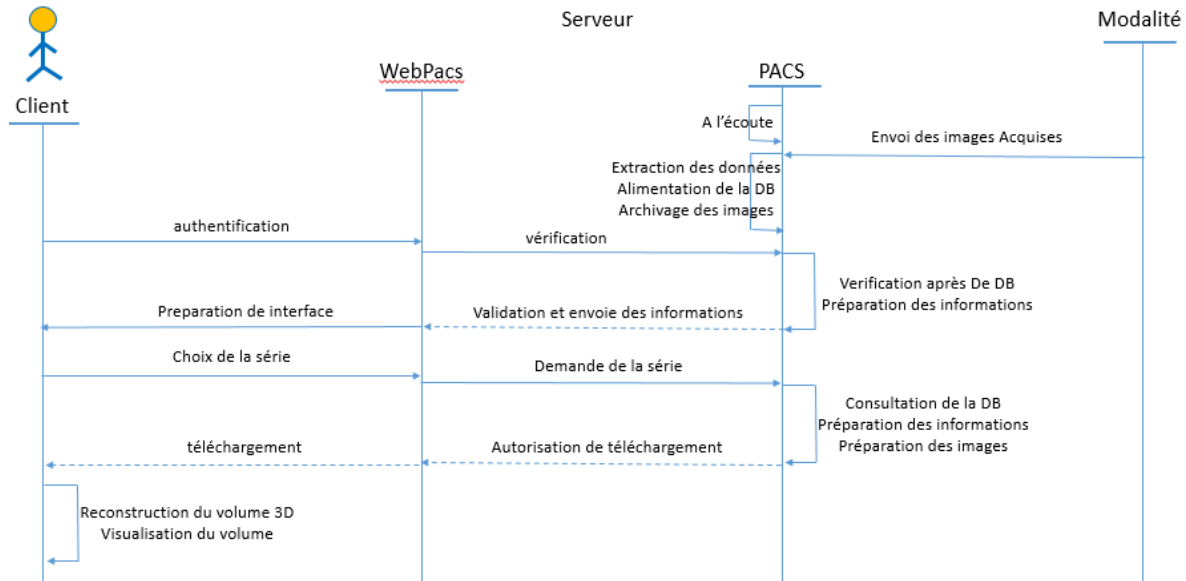
- Parce qu'il répond exactement au problème de plateforme (solution marche pour tous les plateformes).
- Les nouvelles technologies proposées par HTML5 (web API tel que webSocket, webWorker ...) qui renforce la puissance du langage de programmation javascript (plus sa puissance de rapidité de calcul – temps de réponse –).
- La facilité de l'utilisation des pages web (très familiarisées par les gens y compris les médecins).

- 3.2.2 DESCRIPTION DE LA SOLUTION

3.2.2.1 DIAGRAMME DE CAS D'UTILISATION :



3.2.2.2 DIAGRAMME DE SEQUENCE :



3.2.2.3 EXPLICATION DE SCENARIO :

Le serveur PACS (Pecture Archiving and Communication System) est toujours à l'écoute sur le port 104, dans l'attente des séquences des images provenant des chez les modalités (CT-scan, IRM...) lors d'une acquisition.

A la réception le serveur persiste les documents DICOM pour extraire les informations nécessaires pour l'alimentation de la base de données, et ensuite réorganisée la séquence sous une forme bien déterminer pour l'archivée sur le disque dur du serveur.

A ce niveau, le médecin radiologue peut commencer son travail par un authentification auprès de l'interface de communication entre le client et le serveur (webPacs).

Le webPacs répond par présentation une liste des informations sur les séquences archivées sur le serveur pour la session en cours.

Le choix d'une série provoque le webPacs de lancer la demande de préparation de la séquence d'images et les informations nécessaire qui serrant télécharger auprès du client pour commencer la procédure de reconstruction du volume, et lancer la visualisation suivant les différentes approches à savoir :

- Multiplanar reconstruction (MPR).
- Surface rendering (SR).
- Volume rendering (VR).
- MIP, Min-IP (Maximum/Minimum Intensity Projection).

- 3.2.3 LES INFORMATIONS NECESSAIRES A LA RECONSTRUCTION 3D :

- Transfer syntax UID (0002,0010)

Représente un ensemble de règles capable de représenter sans ambiguïté un au plusieurs syntaxes d'encodage. En particulier, il permet de communiquer les entités d'application (modalités \leftrightarrow PACS system), de négocier les techniques de codage communes (byte ordering = little/big endian, compression type, etc. ...).

Transfer Syntax UID	Transfer Syntax Name	Meta Data encoding	Pixel Data encoding
1.2.840.10008.1.2	Implicit VR, Little Endian	Implicit VR, Little Endian	Raw
1.2.840.10008.1.2.1	Explicit VR, Little Endian	Explicit VR, Little Endian	Raw
1.2.840.10008.1.2.2	Explicit VR, Big Endian	Explicit VR, Big Endian	Raw
1.2.840.10008.1.2.4.xx	JPEG xxx (e.g. JPEG 2000)	Explicit VR, Little Endian	JPEG
1.2.840.10008.1.2.4.10x	MPEG xxx (e.g. MPEG2)	Explicit VR, Little Endian	MPEG

- SOP Instance UID (0008,0018)

Représente un identifiant de fichier DICOM, de manière séparée de celui du Study UID, Serie UID. SOP Instance UID est un identifiant unique de l'image DICOM dans la série des images.

- Instance number (0020,0018)

Représente le numéro du fichier DICOM dans la série.

- Slice location (0020,1041)

Représente la position en millimètre de la coupe par rapport à la table de l'acquisition (sert pour la réorganisation des fichiers lors de la reconstruction 3D, et le calcul de la marge pour le processus d'interpolation).

- Slice thickness (0018,0050)

Représente l'épaisseur de la coupe (utilisée pour les calculs réels tel que la distance sur un volume, et aussi elle est utilisée pour le processus de reconstruction et d'interpolation).

- Sample per pixel (0028,0020)

Représente le nombre de plans distincts dans l'image. Un et trois plans d'images sont définis. Autres plans d'images sont autorisés, mais leur signification ne sont pas définies par la norme actuelle.

Pour un type monochrome (échelle de gris) et des images en couleurs de palette, le nombre de plans égale à 1.

Pour le type RGB et d'autres modèles de couleurs de trois vecteurs, cette valeur égale à 3.

- Rows (0028,0010) / Columns (0028,0011)

Représente le nombre de lignes et de colonnes de la matrice de pixels de l'image.

- Pixel spacing (0028,0030)

Représente l'espace occupé par un pixel de l'image, cette valeur contient les dimensions d'un pixel en millimètre suivant x, y.

- Bits allocated (0028,0100)

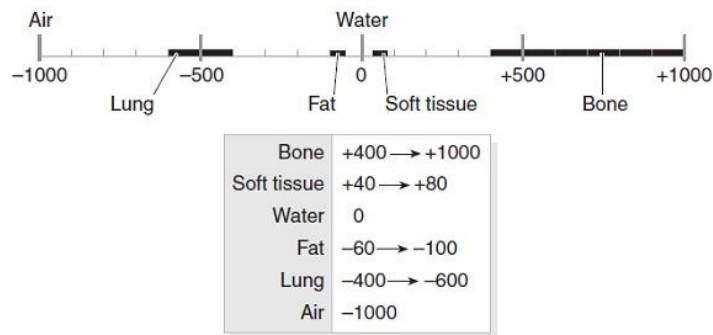
Représente le nombre de bits alloués pour coder chaque pixel.

- Window center (0028,1050) / Window width (0028,1051)

Représente respectivement le centre de la fenêtre et sa largeur, utilisées pour l'application de l'ajustement de contraste de l'image.

- Rescale intercept (0028,1052) / Rescale slope (1052)

Représente les deux valeurs utilisées pour le passage des valeurs stockées (SV) vers la matrice en Hounsfield (HU) – la figure représente l'intervalle des valeurs HU – suivant la formule suivant : $HU = SV * Rescale Slope + Rescale Intercept$



- Pixel Data (0FE0,0010)

Représente un flux de données des valeurs stockées (SV) qui composent l'image. à partir de cette offset en récupère la matrice des pixels.

- 3.2.4 NOTION DE BASE

Avant de voir le détail du processus de reconstruction 3D, il y a certaines notions qu'il faut les connaître.

3.2.4.1 AJUSTEMENT DE CONTRASTE :

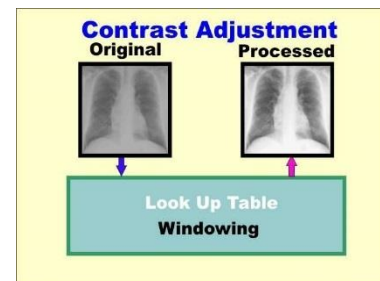
- 3.2.4.1.1 DEFINITION : (WIKIPEDIA)

Le contraste d'une image caractérise la répartition lumineuse d'une image. Visuellement il est possible de l'interpréter comme un étalement de l'histogramme de luminosité de l'image.

- 3.2.4.1.2 METHODES D'AJUSTEMENT DE CONTRASTE :

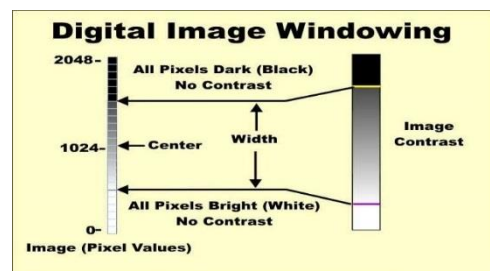
Il existe plusieurs méthodes permettant l'ajustement des caractéristiques de contraste, les deux méthodes les plus utilisées en radiographie numérique sont :

- Lookup table (LUT).
- Windowing (fenêtrage) – méthode adoptée par notre solution –



- 3.2.4.1.3 PRINCIPE D'AJUSTEMENT DE CONTRASTE : (PAR FENETRAGE)

Le fenêtrage est le processus de sélection d'un segment de la gamme totale des valeurs de pixel (gamme dépend de la capacité des récepteurs). Puis représenter les valeurs des pixels à l'intérieur de ce segment de pleine luminosité (nuance de gris) allant du blanc jusqu'au noir.

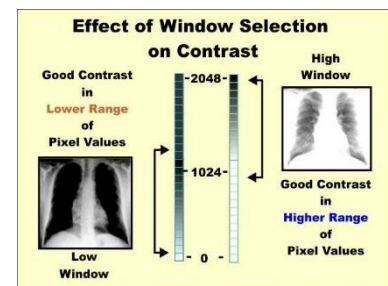


Le contraste sera visible que pour les valeurs de pixel qui se trouvent dans la fenêtre sélectionnée. Toutes les valeurs qui sont en dessus ou en dessous de la fenêtre serrant tous des blancs ou des noirs.

- 3.2.4.1.4 EFFET DE CHANGEMENT DE 'WINDOW LEVEL' :

Avec le principe de fenêtrage, nous pouvons créer de nombreuses images à partir de l'image d'origine par un changement de niveau de fenêtre sur la gamme totale.

Comme nous voyons sur la figure, lorsque la fenêtre est sur le niveau inférieur du totale de la gamme, on a un bon contraste dans les zones claires comme le médistinum. Lorsque la fenêtre est sur le niveau supérieur du totale de la gamme, on a un bon contraste dans les zones sombres comme les poumons.



3.2.4.2 DIFFERENTS APPROCHES DE VISUALISATIONS 3D :

- 3.2.4.2.1 LES APPROCHES VOLUMIQUE VS LES APPROCHES SURFACIQUES :

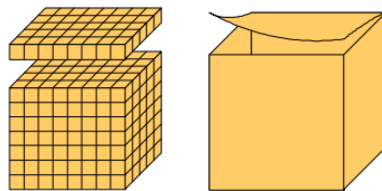
- Le concept :

Le concept de visualisation et de reconstruction se divisent en deux approches duales :

- Techniques dites de rendu de surface
- Techniques dites de rendu de volume.

Généralement, il existe la même dualité entre volumes et surfaces dans les volumes numériques (images numériques 3D) que dans les images numériques 2D entre régions et contours où les régions sont délimitées par des contours.

Représenter un volume à en visualiser la surface. La différence entre les deux approches vient du fait que parfois, on souhaite matérialiser de façon structurée cette surface.

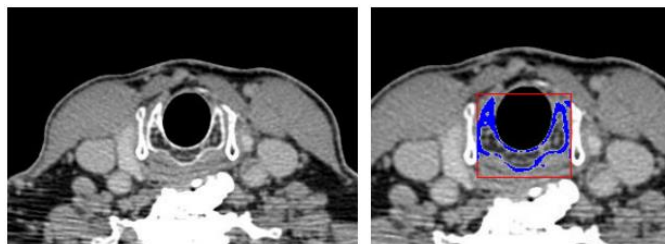


- La segmentation nécessaire :

A la différence de l'image numérique 2D, les volumes numériques doivent être le plus souvent segmentés afin d'en extraire la partie que l'on souhaite étudier : le volume d'intérêt (ou encore les zones d'intérêt).

Différentes techniques peuvent être utilisées :

- Délimitation interactive d'une région est l'approche la plus simpliste et largement utilisée.
- Des seuillages de niveaux de gris peuvent également être mis en place (seuillage manuel, croissance de région, détection de contour, seuillage par détection de vallées, ...).



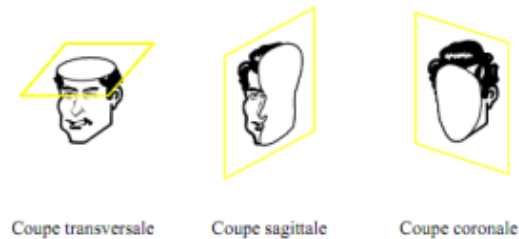
- 3.2.4.2.2 LES APPROCHES VOLUMIQUE VS LES APPROCHES SURFACIQUES :

- La reconstruction planaires (MPR) :

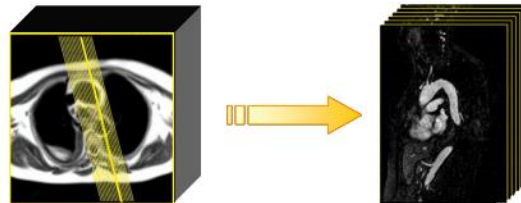
Cette famille de visualisation du volume est la plus simple. Il s'agit de reconstruction d'images 2D, reconstruites à l'intérieur du volume, selon un plan choisi par l'opérateur.

Ce type de reconstruction ne nécessite aucune segmentation particulière.

Les plans les plus standards sont les plans classiques de l'imagerie radiologique : sagittal, coronal et transversal (axial).

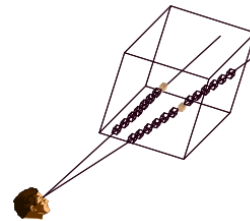


Il existe d'autres plans de visualisations, qui sont les plans courbes (plans oblique, plans double oblique) :

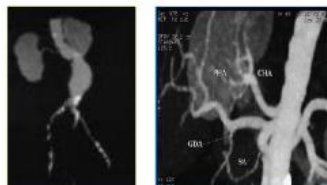


- Maximum Intensity Projection (MIP) :

Le principe est de projeter sur l'image 2D (écran) le maximum d'intensité des pixels rencontrés dans la direction du regard.



Cette technique est devenue quasiment incontournable en imagerie vasculaire où elle est très utilisée, notamment en tomodensitométrie et I.R.M.



Le point fort de cette approche réside dans le fait qu'il s'agit déjà d'une technique regroupement à la fois la segmentation et la visualisation.

La technologie algorithmique employée rappelle quelque peu celle du lancer de rayon. La différence réside dans le critère d'arrêt du rayon qui est choisi ici sur le maximum d'intensité du voxel, sans gérer aucune réflexion. Cette technique donne naissance à une autre technique de même principe mais pour une intensité minimale (Min-IP).

- Les Marching Cubes :

L'algorithme du Marching Cubes a été inventé par Bill LORENSEN et Harvey CLINE. Il s'agit d'une méthode surfacique permettant d'extraire une surface équipotentielle (isosurface) d'un maillage structuré et uniforme 3D.

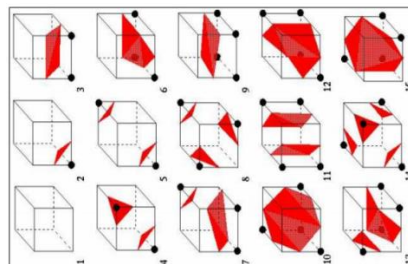
L'objectif de l'algorithme des Marching Cubes est de créer le modèle 3D de la structure anatomique d'intérêt. Afin de représenter un volume, il faut avoir les coupes 2D sériées et segmentées. Ensuite on fixe un seuil de densité et on l'associe à la structure d'intérêt et on divise l'espace en une série de petits cubes élémentaires (appelées voxels).

On distingue donc dans cet algorithme les étapes suivantes :

- Création d'un cube
- Classement des sommets du cube
- Création d'un index pour le cube
- Recherche la configuration correspondante des 15 cas préétablis
- Positionnement du point d'intersection de la surface avec le cube à travers une interpolation linéaire

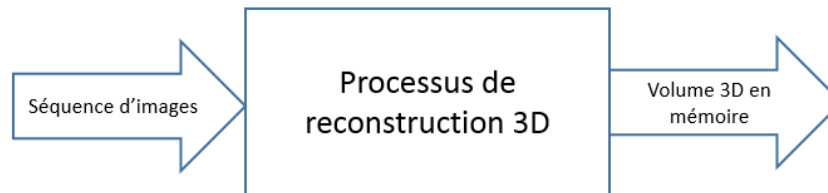
La mise en place de ce procédé repose sur le constat suivant :

- Chaque voxel est un élément cubique. A ce titre, il est composé de 8 sommets, appelés vertex.
- Pour chaque vertex, on doit répondre à la question de savoir de quel côté de la surface il se positionne. Si on considère les 8 vertex d'un voxel, on se retrouve devant 28 soit 256 configurations possibles. Dans la pratique, le jeu des symétries et rotations fait que l'on se retrouve avec 15 configurations distinctes.



3.3 Processus de reconstruction

C'est un processus qui prend en entrée une séquence de fichiers DICOM (obligatoirement de la même série) pour produire comme sortie un volume 3D chargé en mémoire, et prêt à être manipulé (en visualisation, en traitement ...).



- 3.3.1 ETAPES DE RECONSTRUCTION

Les étapes de ce processus sont organisées de la façon suivante :

3.3.1.1 INITIALISATION :

Cette étape consiste à donner une valeur initiale à certaines variables, qui serviront les entrées pour le calcul de volume.

On distingue deux types de paramètres :

- Paramètres initialisés par les données extraites à partir des fichiers DICOM :
 - Rows : nombre de lignes de la matrice des pixels.
 - Columns : nombre de colonnes de la matrice des pixels.
 - Pixel spacing : les dimensions (suivant x, y) d'un pixel.
 - Nombre d'images : nombre de fichiers DICOM de la série.
 - Etc... (généralement ces paramètres sont ceux définis dans la partie 'Les informations nécessaires à la reconstruction 3D').
- Et les paramètres calculés à partir des paramètres précédemment mentionnés :
 - Listes des positions (slice location) par rapport à la table d'acquisition.
 - Premier, dernier plan – position du plan – (first, last).
 - Liste des épaisseurs (slice thickness).

Tous les images ont une position, et une épaisseur. Le calcul et le remplissage des listes et l'initialisation du premier et dernier plan, ce fait à l'aide des paramètres distance space et distance unite et bien d'autres (définis par la suite).

- Distance space

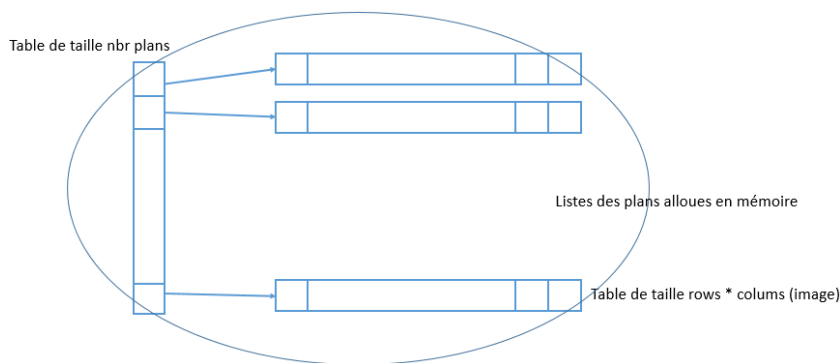
Généralement un pixel est de taille carrée ($x=y$), alors cette donnée sera égale à une des deux dimensions. Ou à un si Pixel spacing n'est pas définis dans le fichier.

- Distance unite

Représente l'épaisseur de la première image (généralement les épaisseurs sont égales). Pour reconstruire un volume avec une bonne qualité de visualisation, on prend généralement un voxel cubique ($x=y=z$). Dans les cas où l'épaisseur est différente de distance space, alors distance unite sera recalculer ou initialiser afin d'avoir des voxels cubique par la suite, ce qui entraîne un changement de nombre de plans.

- Nombre de plans

Nouveau nombre = $E \left[\frac{\text{last} - \text{first} + 1}{\text{distance unite}} \right]$



3.3.1.2 CHARGEMENT DES IMAGES / INTERPOLATION DES PLANS :

Dans cette étape, on cherche à charger les images en mémoire et en même temps à calculer le volume. Mais d'avant de procéder, il faut définir certains paramètres qui servent de calculs lors de cette étape :

- Marge : est la distance à l'échelle réelle, entre la première position du premier plan (first) et la dernière position de dernier plan (last).

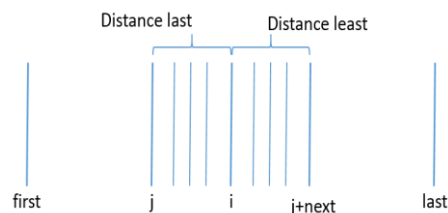
Marge = last – first

- Distance least, distance last :

Soit :

i les positions des nouveaux plans.

j les positions des plans coïncidant avec les images d'origines.



Distance least : est le nombre de plans entre un plan i et le plan j+next, qui le suit directement.

Distance last : est le nombre de plans entre un plan i et le plan j, qui le précède directement.

$$\text{Distance least} = E [i - ((j + \text{next}) - \text{first}) * (\text{nbr plans} - 1) / \text{marge}]$$

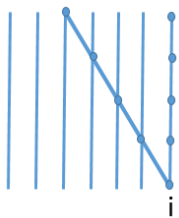
$$\text{Distance last} = E [i - ((j - \text{first}) * (\text{nbr plans} - 1)) / \text{marge}]$$

Ces deux valeurs sont utilisées dans la phase d'interpolation si les images sont orthogonales (l'orthogonalité se détermine à l'aide de IMAGE ORIENTATION (0020,0037) - fichier DICOM -).

Dans le cas contraire ou les images ne sont pas orthogonales, un autre paramètre qui entre en calcul :

- Decalage : permet de calculer la vraie position des pixels d'un plan i

Pour illustrer :



Les pixels du plan i sont les valeurs projetées depuis les autres plans coïncidant avec le facteur d'inclinaison (plan incliné).

Pour calculer cette valeur, il faut déterminer le facteur d'inclinaison qui correspond à la valeur négative de la dernière valeur de l'attribut IMAGE ORIENTATION.

$$\text{Alors : decalage} = E [((\text{nbr plans} - 1 - i) * \text{inclinaison}) + 0,5]$$

Il ne reste que faire de l'interpolation pour charger le volume 3D en mémoire.

- Interpolation :

Le type d'interpolation choisi pour le remplissage des nouveaux plans, est l'interpolation linéaire, suivant les deux formules :

- Si les images sont orthogonales

$$\text{Newbuffer}(\text{indice}) = E [((\text{distance least} * \text{buffer}(\text{indice})) + (\text{distance last} * \text{buffer2}(\text{indice}))) / (\text{distance least} + \text{distance last})]$$

- Sinon

$$\text{Newbuffer}((\text{decalage} + \text{indice}) * \text{columns} + \text{qq}) = E [((\text{distance least} * \text{buffer}(\text{indice} * \text{columns} + \text{qq})) + (\text{distance last} * \text{buffer2}(\text{indice} * \text{columns} + \text{qq}))) / (\text{distance least} + \text{distance last})]$$

On utilise columns pour les coupes axiales, et nbr plans pour les autres.

Avec qq : decalage + indice => rows ou columns.

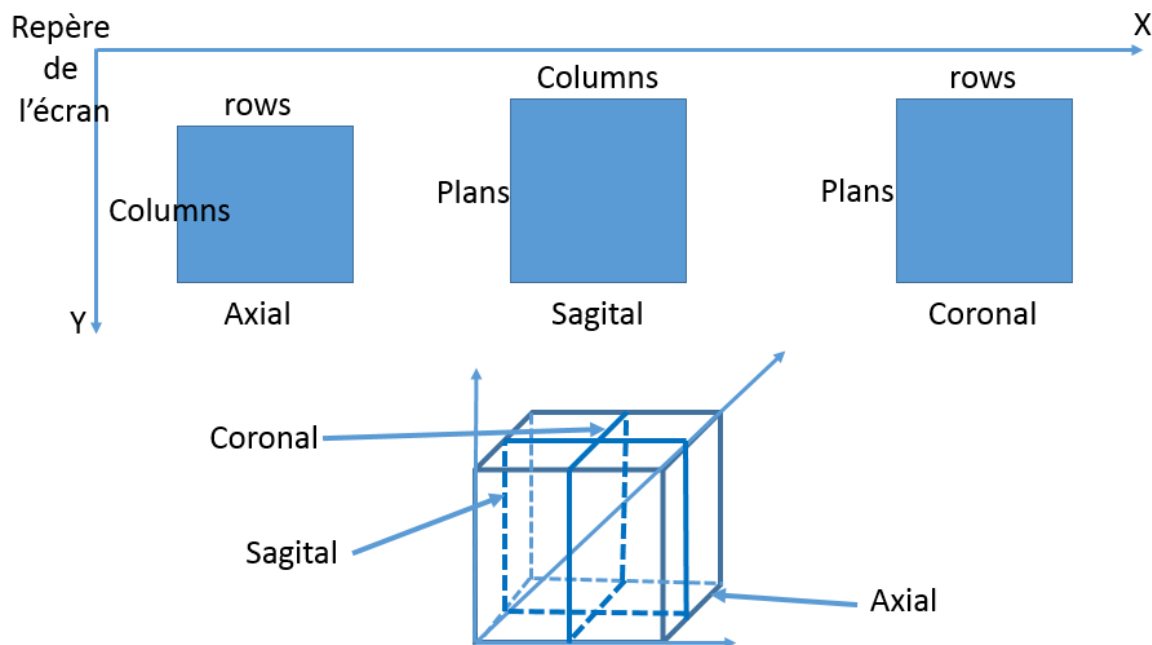
3.3.1.3 VISUALISATION :

Imaginant qu'on a un repère (x,y,z) , sur lequel on a placés les plans axials dans l'ordre l'un sur l'autre (note que la procédure précédente nous a donné les plans dans l'ordre contenant les images d'origines et les nouveaux reconstruites)

- 3.3.1.3.1 LA RECONSTRUCTIONS MPR :

La reconstruction MPR, nécessite la connaissance du point (x,y,z) référence pour déterminer les trois plans (passant par ce point, et qui sont perpendiculaires sur les axes x , y et z) a reconstruite ($\perp X = \text{axial}$, $\perp Y = \text{sagital}$ et $\perp z = \text{coronal}$).

Les plans sont toujours de la forme :



- Reconstruction d'un plan Axial

Pour faire, il faut deux paramètres la coordonnée X du point référence et l'épaisseur de la coupe ($2 \cdot T$) (ces deux paramètres sont spécifiés par l'utilisateur).

Ces deux paramètres permettent de déterminer le plan ou les plans qui participe à la reconstruction d'un seul plan contenant l'image résultante.

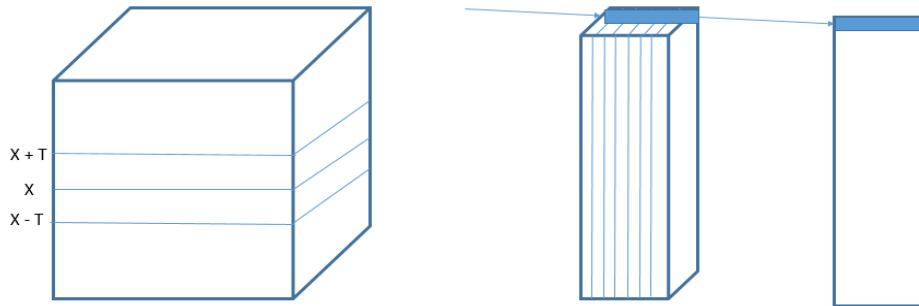
$$\text{Init} = E (\text{Max} (X - T, 0))$$

$$\text{End} = E (\text{Min} (X + T, \text{nbr plans} - 1))$$

Pour illustrer le principe, prenons l'exemple suivant :

$T = 3,$

$X = 200$



La projection se fait selon les approches, vue précédemment tel que MIP (par exemple). Alors sur une position (i,j) de chaque plans entre $X-T$ et $X+T$, on projette celui de l'intensité maximal.

- Reconstruction d'un plan Sagittal ou Coronal

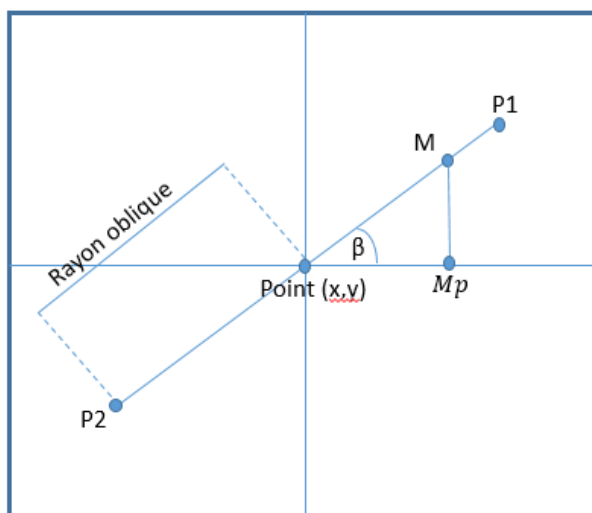
Ce fait de la même manière a la différence l'utilisation du Y ou Z au lieu de X et columns ou rows au lieu de nbr plans.

- Reconstruction d'un plan Oblique

Pour faire, on a besoin de :

- Point (x, y) , est le point MPR sur la coupe à laquelle on veut appliquer la coupe oblique.
- Rayon oblique, égale au maximum valeur entre rows, columns et nbr plans, divisé par 2.

Il faut calculer les extrémités du segment de la coupe



$$\beta = \cos^{-1}(M / \text{dist}(Mp))$$

$$P1 = \begin{cases} x = P1x = \text{rayon oblique} * \cos \beta \\ y = P1y = \text{rayon oblique} * \sin \beta \end{cases}$$

M position de clique de l'utilisateur

De même pour le calcul des coordonnées du point P2.

Par la suite il faut déterminer la fonction de la droite contenant le segment $[P1, P2]$

$F(x, y) = a * P(x, y) + b$ ou a est la pente de la droite, $P(x, y)$ point appartenant $[P1, P2]$

Remarquant ici, si on est sur un plan axial, alors le plan résultant est de taille nbr Plan x k

$$(k = |P1y - P2y|)$$

Pour éviter ce problème de taille de l'image résultante sera toujours 512 x 512, ce qui nous impose de repartir les lignes de l'image sur une matrice de 512 x 512, et de faire une interpolation bicubique (b-spline d'ordre 3) pour remplir le reste à l'aide du noyau de convolution $k(x)$

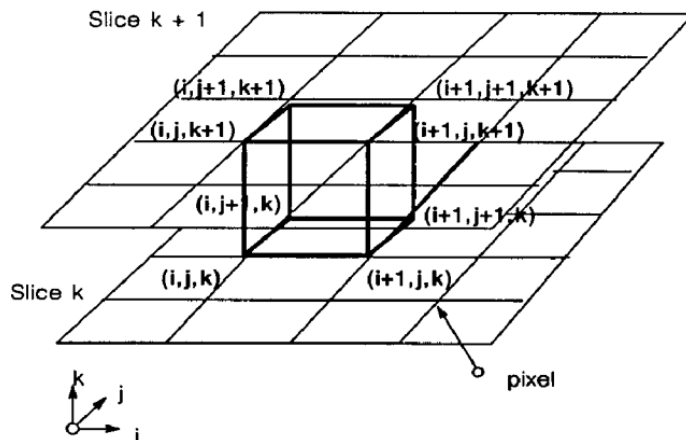
$$K(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & |x| \leq 1 \\ (a)|x|^3 - (5)|x|^2 + (8a)|x| + 4a, & 1 < |x| < 2 \\ 0 & \text{sinon} \end{cases}$$

3.3.1.3.2 VISUALISATION MARCHING CUBES :

- Les Etapes à suivre
 - Création d'un cube
 - Classement des sommets du cube
 - Création d'un index pour le cube
 - Recherche la configuration correspondante des 15 cas préétablis
 - Positionnement du point d'intersection de la surface avec le cube à travers une interpolation linéaire

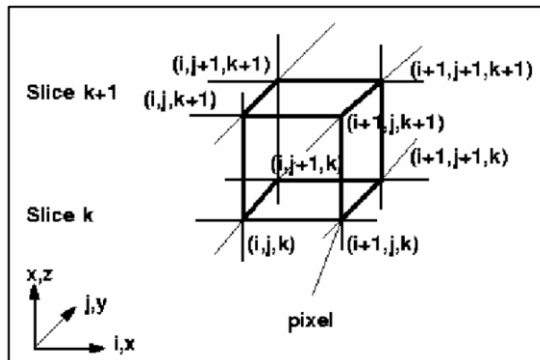
Cet algorithme commence par placer les plans l'une sur l'autre pour faire la reconstruction surfacique de l'objet

Pour illustrer ce mécanisme, prennent seulement deux plans :

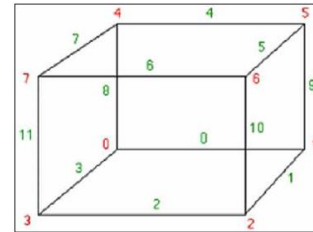


Deux images 2D adjacentes fournissent les 8 pixels formant un cube (comme sur la figure). Pour localiser la surface dans ce cube logique, on applique l'algorithme de marching cubes qui permet de déterminer l'intersection de la surface avec ce cube puis passer au cube suivant.

Soit un cube défini par 8 sommets, est l'intersection de deux tranches adjacentes, 4 de la tranche k et les Autres de la tranches $K+1$.



Pour la numérotation des sommets et des arêtes d'un cube on suit la convention Paul Bourke [BOU 94].

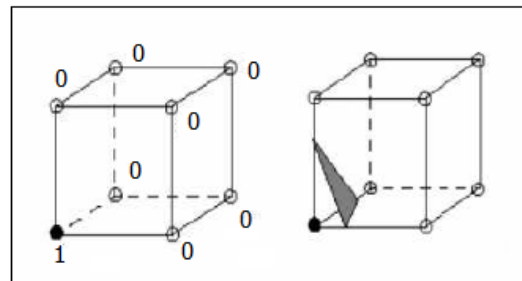


- Détermination de l'index

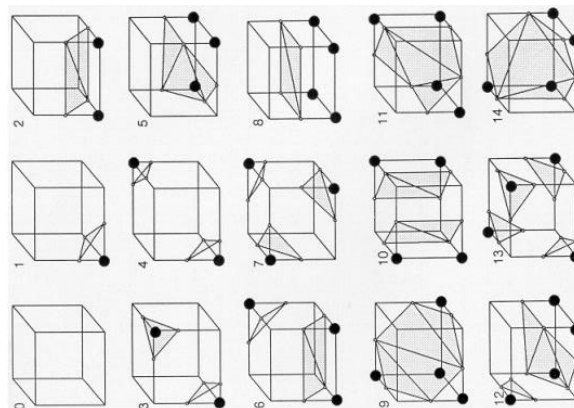
Pour la création d'un index on doit classifier les sommets des cubes par :

- 1 s'il est à l'intérieur de la surface.
- 0 s'il est dehors.

La surface coupe alors les arêtes du cube lorsque les 2 vertex formant l'arête sont de signes contraires.



Par la suite on doit créer une topologie de la surface dans le cube, et puisqu'il y a 8 sommet dans le cube et chaque sommet est représentée par deux états binaires 0 et 1, on a alors 256 cas des surfaces qui peut se croiser dans un seul cube ou 256 configurations possibles de répartition des sommets, mais grâce à des symétries de rotation et d'inversion des points internes/externes, on peut se ramener à 15 configurations seulement qui donnent 15 topologies différentes de triangles, représentées dans la figure suivante :



- Détermination des intersections

On pourra reconnaître ces modèles ou l'ensemble des intersections sur les arrêtes des cubes à partir d'un index crée en fonction de leurs sommets.

L'index sera un pointeur dans l'Edge Table [PEY 03] qui est une table intermédiaire de 256 éléments. Les entrées de Chaque élément de la table font 16 bits :

- 12 bits sont utilisés,
- 4 bits inutilisés pour des raisons de commodité et de performance

Cette table permet de savoir quels points sont à calculer grâce à l'interpolation linéaire et évite ainsi les calculs inutiles.

Par exemple si dans un cube les sommets 0 et 2 appartiennent à la surface alors l'index est : 00000101 sa conversion dans la base 10 est égale à 5 donc cet index correspond au 5ième élément de l'edge table.

Edge Table= [0x000, 0x109, 0x203, 0x30a, 0x406, 0x50f, ...0x109, 0x000];

0x50f =

0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1
/	/	/	/	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

D'où les arrête : 0, 1, 2, 3, 8 et 10 sont coupées par la surface.

- Interpolation de l'emplacement des points d'intersections

Pour chaque pointe du triangle, on cherche l'emplacement du sommet sur l'arrête par l'interpolation linéaire des valeurs du voxel.

$$X = i + \left[\frac{T - V[i]}{V[i+1] - V[i]} \right]$$

T : est le iso-value ou la valeur de la densité.

i : est le numéro du sommet de l'arrête intersectée

V[i] : est la valeur scalaire correspond à chaque sommet.

- 3.3.2 PROBLEME DE RESSOURCES

Lors de développement de la solution, on s'est trouvé face à certains problème qu'ont donnée fin à cette solution et naissance à une autre.

3.3.2.1 LIMITATION DE MEMOIRE :

Pour travailler avec un navigateur, on est obligé d'être limiter par certains critères (sont gères par le créateur du navigateur) pour chaque processus (pour un navigateur un processus est un anglet) démarré par un navigateur. Si je prends l'exemple de Google chrome (outil principalement utilisée pendant le développement), chaque processus ne peut utiliser que 2GO de mémoire (RAM) en totale, et il faut utiliser 1,5GO pour le sauvegarde des données et laisser 0.5GO pour le traitement (appelle de fonctions, utilisation de variables locales, gestion des évènements ...). Et pour l'utilisation du CPU, il ne faut jamais dépasser les 512Mbits/s. ce qui rend la tâche impossible pour les séquence d'images qui dépassent les 600 images/série (a noté qu'il existe des séries qui peuvent dépasser 2000 images).

3.3.2.2 COMMUNICATION DES DONNEES ENTRE THREADS (WEB WORKER) :

Pour illustrer ce problème, supposant qu'on a un thread principal et son thread fils.

La communication se fait uniquement par postMessage, cette communication est basée principalement sur le principe de clonage de données, c'est-à-dire :

- Le thread principal commence par préparer une copie pour être transférer via postMessage.
- Le thread fils réserve de la mémoire pour recevoir la copie.

Ce qui veut dire que la taille de la mémoire allouée sera multipliée par deux. Et plus précisément, a un moment donnée la taille sera multipliée par trois. Et quand il s'agit une taille volumineux cette opération prend énormément de temps parce qu'il utilise le mécanisme de sérialisation. De plus javascript ne permet de supprimer et de libérer de la mémoire de manière manuelle, c'est garbage collector qui s'en occupe. A savoir qu'on utilise huit thread fils pour le calcul de volume, pour rendre le temps de réponse minimale. Alors pour résoudre le problème de temps de PostMessage, on a pensé subdiviser les données sur huit, mais un autre problème apparait concernant la reconstruction 3D (l'interpolation, lancer rayon ...).

La seule moyenne, pour résoudre tous ces problèmes, est de forcer les threads fils d'utiliser la mémoire allouée par le thread principal en mode lecture et écriture (ce qui n'est pas possible par l'api WEB WORKER de HTML5 pour des raisons de sécurité).

Ce qui nous amené vers la fin du chemin pour cette solution qui consiste à faire tout le calcul auprès du client. Et de penser à faire le calcul auprès d'un serveur et n'affiche que des images résultantes auprès du client (bien sûre quelques calculs sont indispensables).



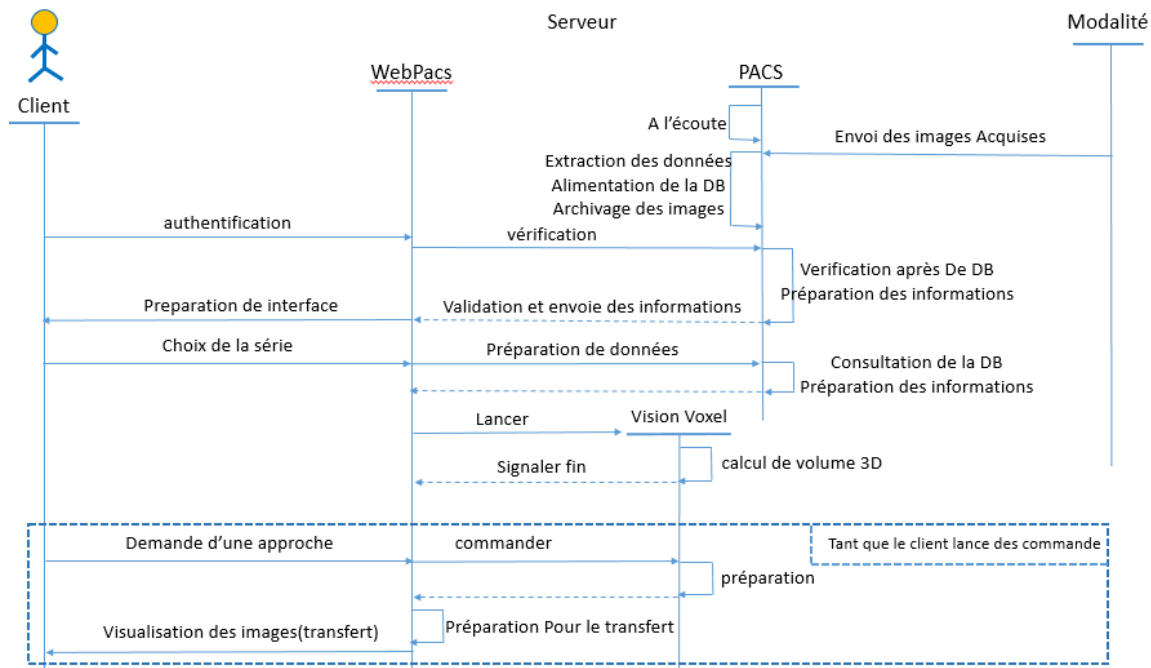
3.4 Présentation de la deuxième solution

Le nouveau chemin propose une solution aux problèmes rencontrés durant la réalisation de la première solution

- 3.4.1 DESCRIPTION DE LA SOLUTION

3.4.1.1 DIAGRAMME DE CAS D'UTILISATION (TOUJOURS LE MEME) :

3.4.1.2 DIAGRAMME DE SEQUENCE :



3.4.1.3 EXPLICATION :

On ait toujours dans le cadre de la réalisation de l'outil qui répond au besoin précédemment présenter. A la seule différence que le traitement sera subdiviser en trois parties :

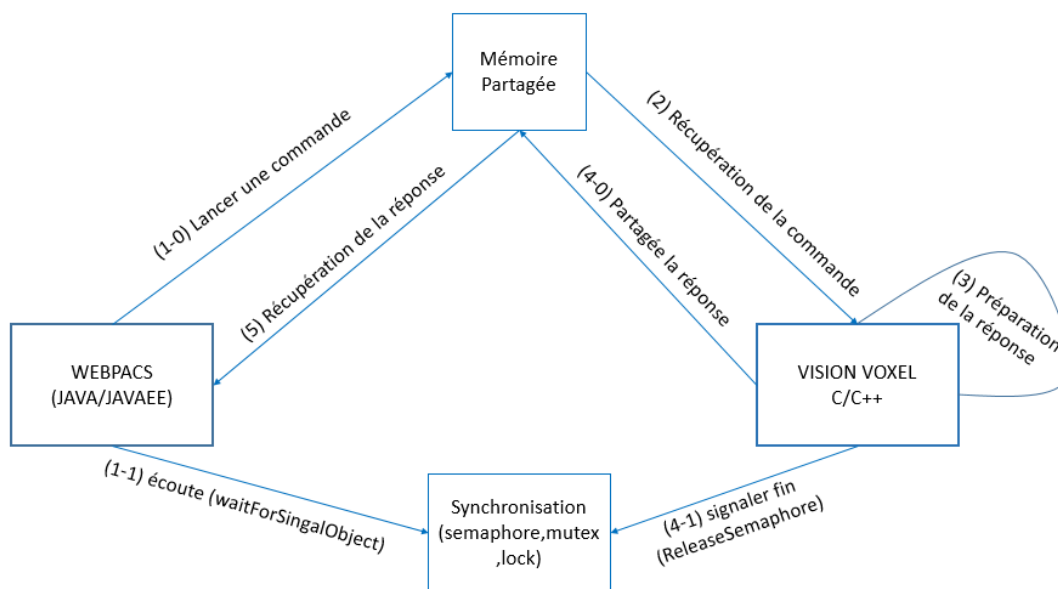
- Partie une (serveur) : calcul et chargement du volume dans la mémoire.
- Partie deux (serveur) : reconstruction d'image 2D et compression pour le transfert.
- Partie trois (client) : visualisation et design des différentes figures sur les images.

Pour réaliser cette solution, on se serve de la puissance de calcul et de temps de réponse du langage C/C++ (VISION VOXEL), la sécurité et la gestion des applications proposer par la technologie JavaEE, et bien sûr HTML5/Javascript sont toujours présent.

VISION VOXEL s'occupe de la reconstruction du volume et de répondre aux demandes des clients à travers WEBPACS.

WEBPACS s'occupe de reproduire l'image (réponse de VISION VOXEL) sur la mémoire sous format JPEG (compressée avec une qualité réglée) et le transfert de l'image au client.

La figure suivante, illustre la technique de communication entre les deux entités :



Cette technique nécessite :

- L'utilisation de la mémoire partagée pour faire communiquer les deux entités.
- L'utilisation des techniques semaphore, mutex et lock pour assurer la synchronisation de traitement entre les deux entités et aussi entre les inter-threads.

3.4.1.4 PROBLEME DE TRANSFERT :

Vue qu'on cherche à préparer des images pour les transférer par le réseau (plusieurs transferts par seconde), on sera pénalisé par ce temps de transfère pour les différentes bondes passantes. Ce qui nous impose de suivre la Stratégie d'optimisation de temps de traitement sur serveur :

- Stratégie Syntaxe : optimisation de code source (par exemple : déclaration de minimum de variable, minimiser le maximum possible les calculs...).
- Stratégie fonctionnelle : dépend des fonctionnements de gestion

Je prends l'exemple de la gestion des requêtes http par le serveur web TOMCAT :

La gestion se fait par fil d'attente et par quota, donc je dois effectuer des requêtes qui serrant réaliser avec la fin de quota par exemple.

Le tableau suivant montre les délais des traitements pour les différentes bondes passantes (Temps de téléchargement / temps de reconstruction sur serveur en ms) :

Taille d'image		512 x 512	512 x 512	512 x 512
Nombre de plans		322	811	1472
Good 2G	Axial	537 / 25	1205 / 14	434 / 23
	Sagital	563 / 20	1549 / 33	730 / 40
	Coronal	695 / 10	1501 / 30	905 / 33
Regular 3G	Axial	315 / 13	320 / 13	274 / 23
	Sagital	351 / 20	484 / 31	501 / 44
	Coronal	428 / 9	498 / 19	640 / 44
Good 3G	Axial	156 / 23	146 / 14	109 / 13
	Sagital	174 / 20	421 / 36	200 / 45
	Coronal	202 / 10	428 / 20	279 / 33
Regular 4G	Axial	62 / 13	62 / 13	45 / 23
	Sagital	86 / 11	170 / 32	93 / 45
	Coronal	93 / 10	176 / 20	107 / 32
DSL	Axial	92 / 13	86 / 16	59 / 26
	Sagital	90 / 11	286 / 32	135 / 41
	Coronal	126 / 9	292 / 31	198 / 42
Wifi	Axial	14 / 14	12 / 24	11 / 13
	Sagital	14 / 21	22 / 36	26 / 41
	Coronal	18 / 9	25 / 20	28 / 32
Réseau local	Axial	6 / 23	6 / 38	6 / 14
	Sagital	8 / 20	11 / 31	7 / 24
	Coronal	7 / 10	12 / 41	13 / 22

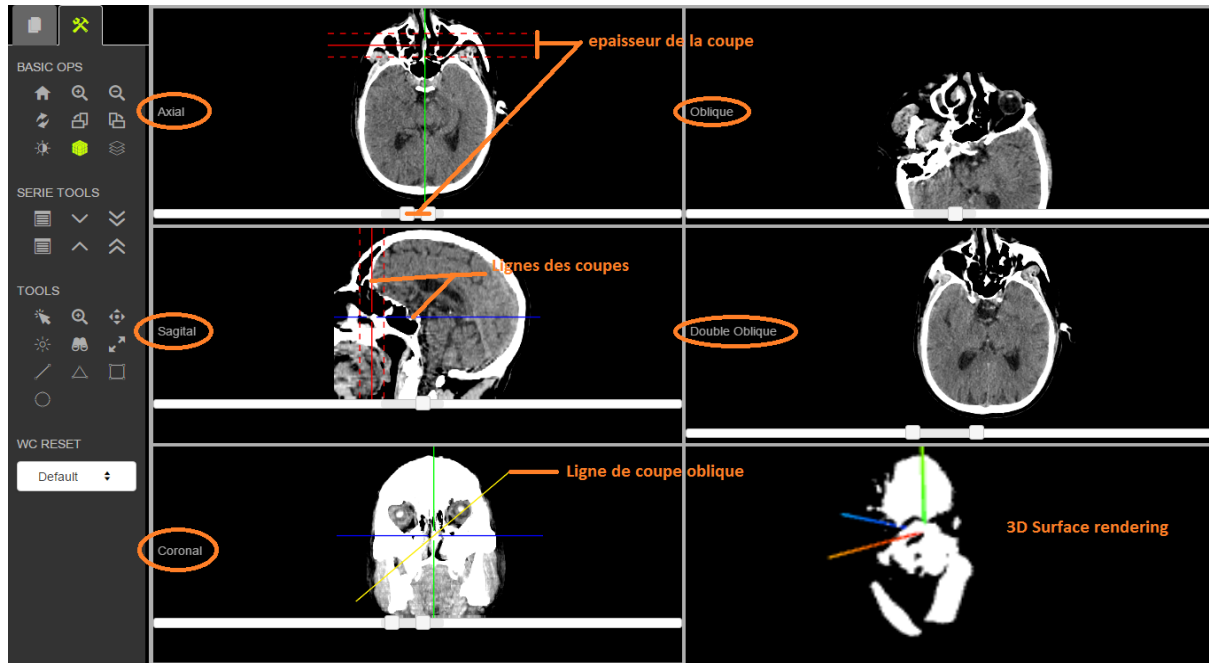
Les prétests sur les délais montre qu'on est le premier par rapport aux autres Software au niveau de temps de réponse :

- INFINITT web viewer (société japonaise classer Premier mondialement).
- MedPac.
- Siemens Syngo via.

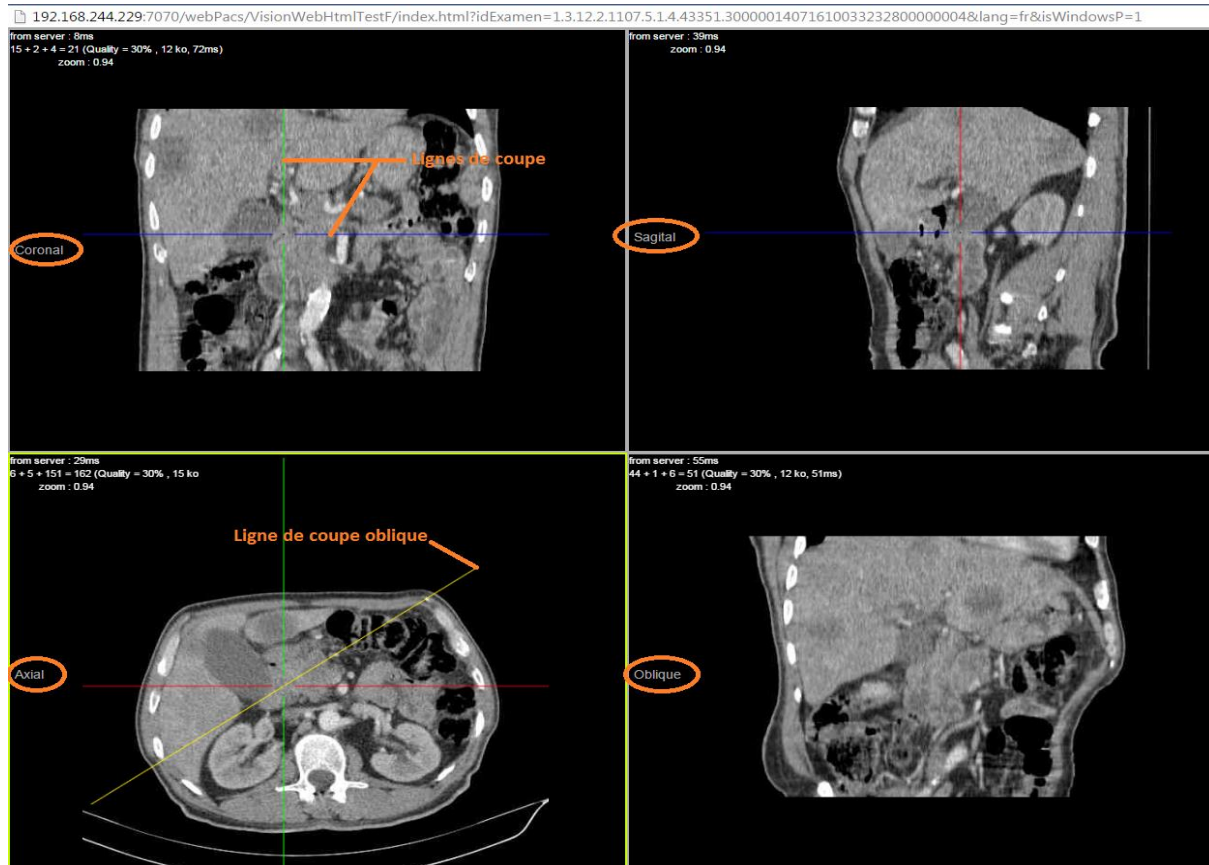
N.B : leurs inconvenant, est qu'ils font tous les traitements sur le serveur, alors que notre solution subdivise le traitement sur trois parties sachant que le test est fait sur un serveur de configuration minimal (CPU : core duo 2.93 GHZ, RAM : 4GO).

- 3.4.2 CAPTURE D'ECRAN

SOLUTION N° 1 :



SOLUTION N° 2 :



Conclusion

Durant les deux années d'études que j'ai passées au sein de la faculté des sciences et techniques, j'ai eu l'opportunité d'acquérir des connaissances au niveau technique, analytique et relationnel, grâce à une direction qui veille sur la qualité de l'enseignement et aux professeurs qui m'ont fait partager leurs connaissances sans modération.

Pour compléter mon cursus, j'étais amené à réaliser un projet de fin d'étude au sein de la société BINARIOS, qui consistait à mener une étude fonctionnelle pour le 'Développement d'une plateforme VISIONWEB pour la reconstruction et la visualisation 3D d'une séquence d'images médicales'. L'objectif ultime étant la réponse au besoin des médecins radiologues et l'amélioration du temps fait pour le diagnostic médical.

Ce projet m'a permis d'enrichir mes connaissances dans des domaines variés comme : la radiographie, l'infographie, les langages C/C++, HTML5/javascript, UML, et la technologie JavaEE, ...

Comme perspective à ce projet, je souhaite l'adoption de mon approche comme solution élémentaire, ce qui sera la récompense idéale pour tout l'effort fourni durant ce projet. Pour objectif de lancer la concurrence aux grandes sociétés de l'informatique médicale, par la présentation d'une plateforme de visualisation 3D qui ouvre au future, les chemins vers des améliorations pour faire du diagnostic automatique et de détection automatique des tumeurs et bien d'autre possibilités....



Références

Site web

- Digital Imaging and communication in medicin
(<http://dicom.nema.org/>)
- Introduction à la norme DICOM
(<http://www.creatis.insa-lyon.fr/~dsarrut/mybib/2004/rapportSeda2004.pdf>)
- Digital Image Processing
(<http://www.sprawls.org/resources/DIGPROCESS/module.htm>)
- Imagerie 3D en Tomodensitométrie
(http://www.utc.fr/~farges/master_mts/2004-2005/projets/reconstruction3d/trois.html)
- DIGITAL IMAGE INTERPOLATION
(<http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>)
- MRI images
(<http://www.mr-tip.com/serv1.php?type=img&img=Brain%20MRI%20Images%20Axial%20T2>)
- Volume Rendering with Higher - Order Interpolation -
(<http://www.vis.uni-stuttgart.de/~bachthsn/projects/highorder/highorder.html>)
- Scientific Visualization
(http://www.cs.utah.edu/~ramanuja/sci_vis/prj4/REPORT.html)
- DICOM le standard pour l'imagerie médicale
(https://www.researchgate.net/publication/247272130_DICOM_le_standard_pour...)

Article

- Imagerie médicale 3D visualisations, segmentation et reconstruction [Bruno NAZARIAN – CNRS - 2002]
- Processive volume rendering using WebGL and HTML5 [University of Dublin, Trinity College - 2011]
- Reconstruction 3D des images médicales par l'utilisation de l'algorithme marching cubes [SETIT 2009]
- Numerical methodes for isosurfaces volume rendering [VIRGINIA TECH, SPRING 2009]
- 3D reconstruction and volume computing in medical imaging [Technical University of Cluj-Napoca 2011]
- 3D DICOM images Reconstruction using LABVIEW [IEEE ISSN: 2320-2084]
- Direct volume rendering [CMPT 467/767]
- Marching cubes : a higher resolution 3D surface construction algorithm [Computer Graphics 1987]

