

## 1. Introduction

L'objectif de ce chapitre est d'établir un état de l'art en matière de rendu volumique. Nous commençons tout d'abord par expliquer ce que sont les données volumiques, puis nous exposons un éventail des méthodes permettant la visualisation de telles données, et nous terminons par décrire certaines améliorations, permettant à la fois d'améliorer l'interactivité et la qualité du rendu.

## 2. Données volumiques

Les données volumiques sont des données multi-dimensions, elles ont des significations variées suivant le domaine d'application concerné. Généralement correspondent à des informations portées par un maillage tridimensionnel.

Il existe différents types de représentation de ces données [1], Elles peuvent être représentées par:

- **Champ scalaire:**

C'est une quantité qui peut être exprimé par un nombre unique représentant de son ampleur (la masse, la densité, la température). La région est appelée un champ scalaire si à chaque point dans une région est associée par une valeur scalaire.

L'exemple suivant représente la température selon les villes de France [1].

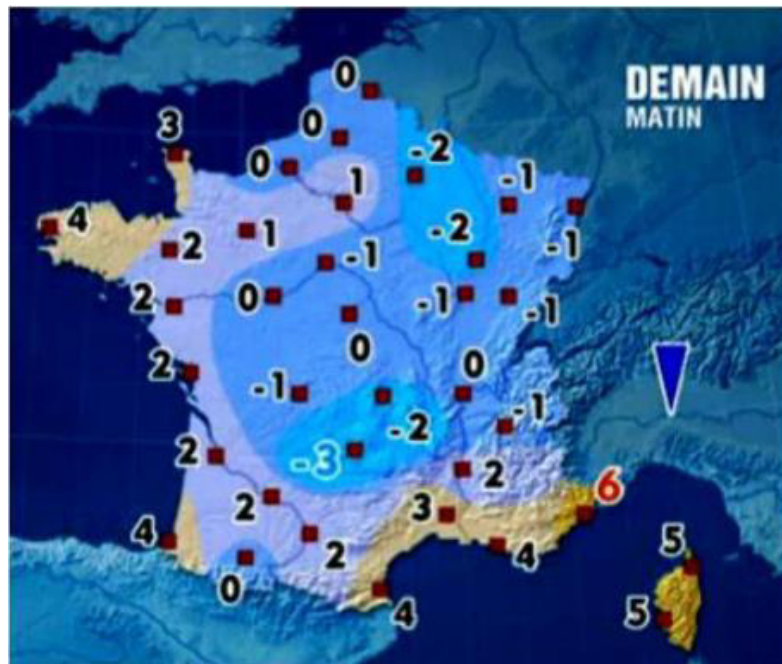


Figure 1.1 Approche graphique de la notion de champ scalaire.



D'abord La notion de structure est principalement liée à la connexité dans le maillage, si la connexité des cellules est implicite, les données sont dites « structurées»

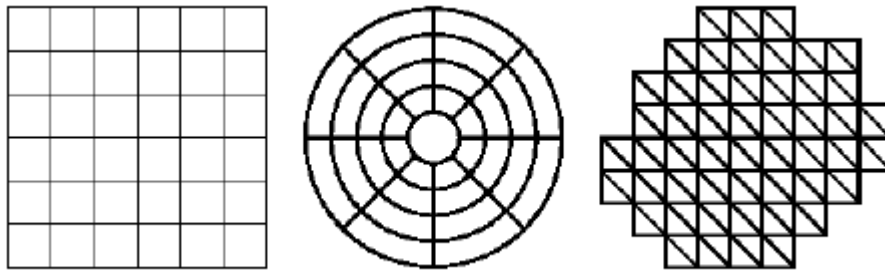


Figure 1.3 Exemples de maillages structurés en deux dimensions [13].

Ainsi Si la connexité entre cellules ne peut être définie qu'explicitement, les données sont alors dites "non structurées"

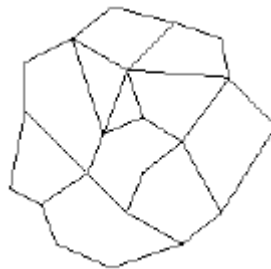


Figure 1.4 Exemple de maillage non structurés en deux dimensions [13].

### 3. La visualisation scientifique

La visualisation scientifique consiste à réaliser l'analyse et l'affichage d'informations pertinentes par rapport à un ensemble de données. Cela peut aller d'une représentation sous forme de courbes, de graphes, d'images, ou de rendu 3D. La visualisation scientifique s'applique à des domaines larges d'activités tels que la visualisation de données : Médicales (données 2D, 3D, ainsi que temporelles), Biologiques, Mécaniques... etc.

Nous avons intéressé dans ce travail par la visualisation volumiques.

Le volume ici est un tableau 3D régulière de valeurs de données, des voxels.

La grille régulière tridimensionnelle peut être considérée comme un empilement de matrices bidimensionnelles de valeurs de données où chacun est une image, et chacune des valeurs de données est représenté par un pixel.

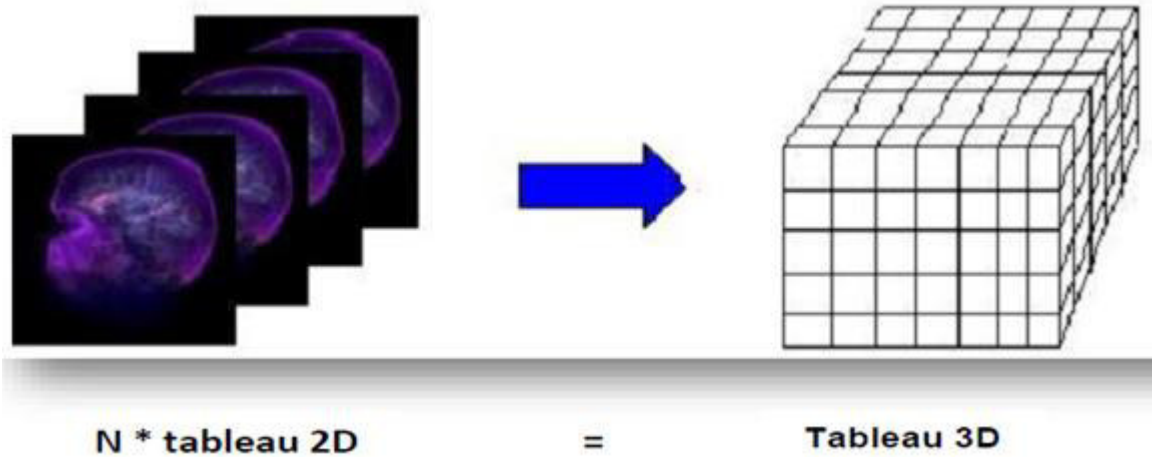


Figure 1.5 Représentation de données volumiques.

Un voxel peut être défini par une seule valeur unique dans toute la région cubique ou un point de données à chaque coin de la cellule cubique[3], La valeur d'un point à l'intérieur de la cellule est déterminée par l'interpolation[4].

## 4. Méthodes de visualisation volumique

La visualisation des données volumiques est le processus de transformation de la nature discrète et complexe de ces données en images comprises par la plupart des utilisateurs dans le terrain médical.

Trois principales méthodes permettent de visualiser des données volumiques:

### 4.1. Visualisation de tranches 2D

Cette méthode consiste à extraire une tranche arbitraire de données, et à la visualiser. On peut comparer cette approche à celle des médecins qui examinent des images obtenues par résonance magnétique (IRM). Ils disposent d'un certain nombre de clichés correspondant directement à des "tranches" successives d'un corps humain. [5]

### 4.2. Visualisation d'isosurfaces

Avec cette seconde méthode, on cherche à reconstruire la (les) surface(s) composant les bords de l'objet en se basant sur la notion d'isosurface, définie comme le lieu des points de l'espace où la fonction scalaire est constante. Un algorithme typique appartenant à cette classe est l'algorithme des "marching cubes" de Lorensen et Cline. [5]

### 4.3. Rendu volumique direct (DVR pour Direct Volume Rendering)

La dernière méthode consiste à visualiser le volume en utilisant l'ensemble des données volumiques. On considère généralement le volume comme un corps participatif auquel on va associer des propriétés optiques, notamment de transparence. C'est sur cette dernière approche que nous nous concentrons, car c'est celle qui permet d'obtenir une vraie vue d'ensemble des données.

## 5. Rendu volumique direct(RVD)

### 5.1. Principe

Les algorithmes de rendu volumique direct utilisent les données originelles du volume. Chaque pixel de l'image est considéré comme un rayon qui traverse un volume "figure 1.6".

La complexité de l'algorithme réside ici dans la manière dont le rayon va interagir avec les échantillons (ou voxels) du volume. Ainsi, tous les algorithmes de rendu volumique direct fonctionnent en échantillonnant des valeurs le long du rayon et en calculant la contribution de ces valeurs, suivant un modèle de rendu, à la couleur finale du pixel [6].

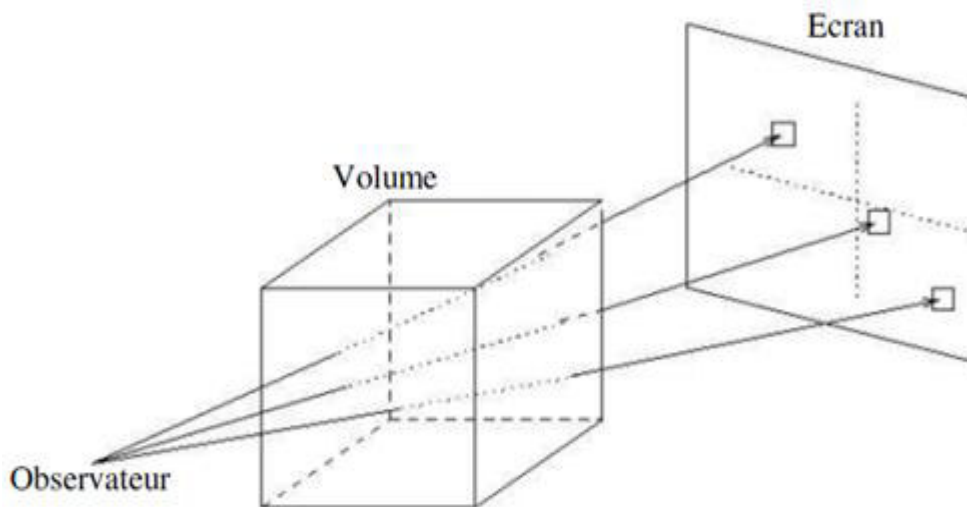


Figure 1.6 Rendu volumique direct (RVD) [14].

Il existe en fait deux manières possibles de calculer l'image finale. La première consiste à calculer la contribution de chaque élément du volume sur les pixels de l'image. Cette façon de calculer l'image est appelée Object-order (figure 1.7) car les voxels sont passés en revue un par un dans un ordre de visibilité donné [7].

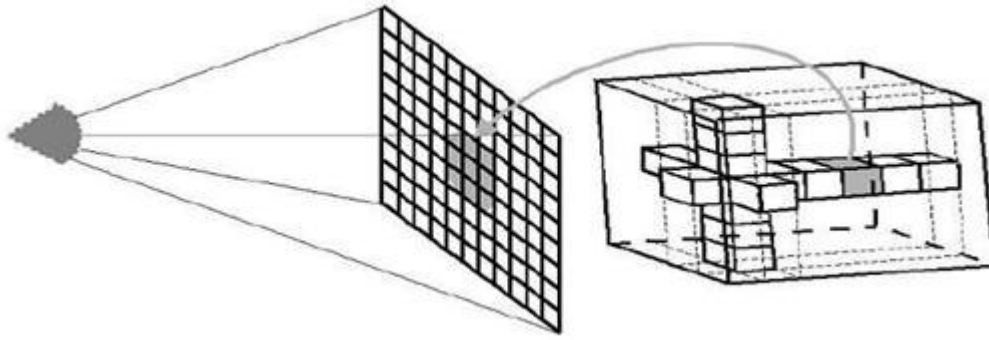


Figure1.7 Les méthodes Object-order

A l'inverse, la seconde manière consiste à calculer chacun des pixels de l'image un par un. Il faut alors que chaque rayon traverse le volume afin de trouver les voxels qui contribuent à la couleur finale. On peut ainsi éviter de parcourir les parties cachées du volume. Cette méthode est appelée **image-order** (figure 1.8). [7]

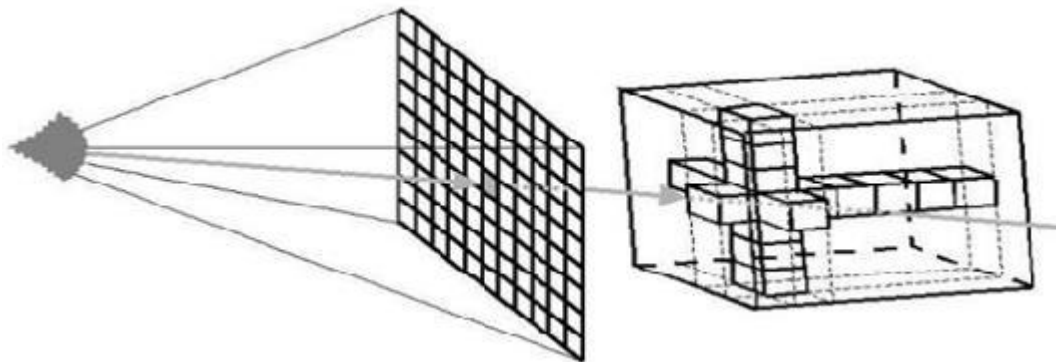


Figure1.8 Les méthodes image-order

## 5.2. Ray Casting

Le **Raycasting** est une technique de rendu volumique direct qui utilise la méthode **image-order**. Le but de Raycasting dans le rendu volumique est de prendre en compte la transparence et la couleur de chaque point. Il consiste à émettre des rayons à partir du point de vue jusqu'au plan de l'image pour chaque pixel en traversant le volume. Lorsqu'un rayon traverse le volume, ce dernier est échantillonné selon un certain nombre d'intervalles, en général, un ou deux par voxel. La valeur à chaque point d'échantillonnage est obtenue par une interpolation tri linéaire des valeurs voisines de ce point. Les contributions de ces échantillons sont accumulées jusqu'à ce que le rayon ait quitté le volume. La valeur finale est alors calculée suivant la formule d'accumulation et placée sur le pixel que le rayon atteint [5].

L'avantage principal de cet algorithme est la qualité du rendu, Cependant, il n'accède pas aux données du volume dans l'ordre naturel du stockage, puisqu'il le traverse dans des directions



arbitraires, ainsi l'élimination des régions vides du volume, bien que faisable, est moins facile qu'avec des méthodes de type object-order, et donc plus coûteuse.

De fait, il passe plus de temps à calculer les positions des points d'échantillonnage et à effectuer les calculs d'adresses qu'à calculer le rendu. De plus, les rayons peuvent passer plusieurs fois par le même voxel, et ainsi il est nécessaire de recharger des données déjà chargées précédemment [5].

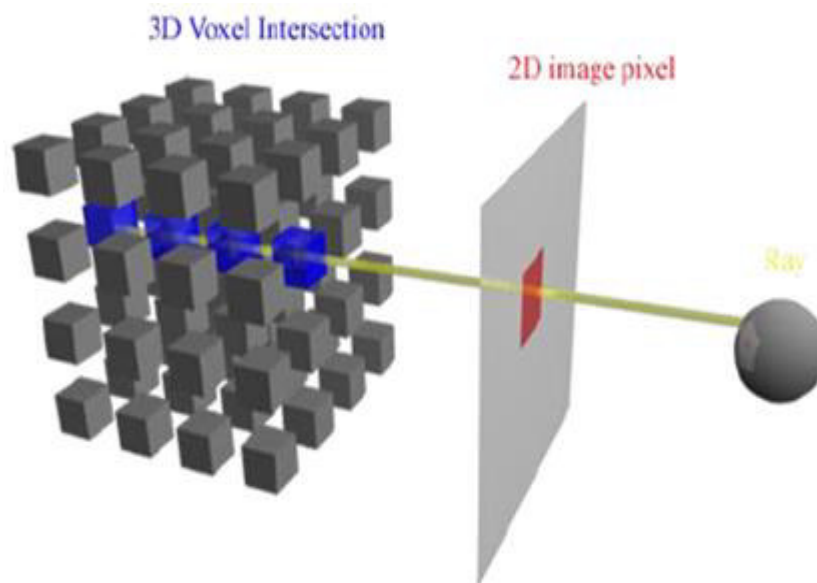


Figure1.9 Le principe de Raycasting

## 6. Fonction de transfert et classification

Le rendu volumique cherche à explorer les données volumiques à l'aide des outils visuels. Ce processus a pour objectif de découvrir et de mettre en évidence des structures et des phénomènes intéressants incorporés dans les données en écartant les structures d'occultation qui ne sont pas actuellement importantes. Les données volumiques sont définies par un maillage portant des valeurs scalaires, il est alors important de pouvoir attribuer des propriétés optiques à ce maillage, afin de pouvoir visualiser les informations qu'il porte. Donc, le rendu volumique nécessite une étape dite la classification qui consiste à transformer l'ensemble de voxels en des voxels visualisables c.à.d. à spécifier la relation entre les données scalaires et les caractéristiques optiques (absorption et émission).

- **Fonction de transfert**

Comme nous l'avons vu ci-dessus, les données volumiques sont définies par un maillage portant des valeurs scalaires. Il est alors important de pouvoir attribuer des propriétés optiques à ce maillage, afin de pouvoir visualiser les informations portées. La fonction de transfert est l'outil qui permet de réaliser cette opération en associant à une valeur scalaire  $s$ , portée par le maillage, un quadruplet  $(r, g, b, a)$  pour rouge, vert, bleu, alpha. Le triplet  $(r, g, b)$  est choisi, car il correspond à l'espace de couleur additif classiquement utilisé en informatique. Le dernier élément permet, quand à lui, d'associer une notion de transparence/opacité aux scalaires  $s$  [5].

Donc classification en rendu volumique est l'étape qui permet de passer de l'espace des scalaires à l'espace des couleurs et de transparence par la définition d'une fonction de transfert

- **classification**

La classification en rendu volumique est l'étape qui permet de passer de l'espace des scalaires à l'espace des couleurs et de transparence par la définition d'une fonction de transfert.

L'ordre d'application de l'interpolation et de la fonction de transfert donne ainsi lieu à deux classifications différentes (**figure 1.10**), car ces deux opérations ne commutent pas (à moins que la fonction de transfert soit une fonction constante ou identité), Autrement, à partir d'une fonction de transfert, il est possible de traiter les échantillons voxels de deux façons [5] :

- la pré-classification :  $s \rightarrow (r, g, b, a) \rightarrow$  interpolation.
- a post-classification :  $s \rightarrow$  interpolation  $\rightarrow (r, g, b, a)$ .

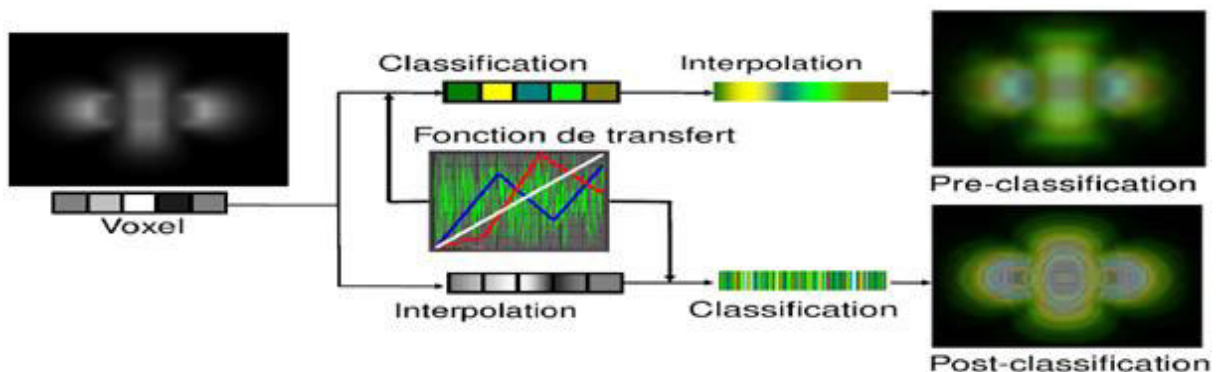


Figure 1.10 Classification



Les deux méthodes disposent chacune d'avantages et d'inconvénients. Du fait que la pré-classification réalise l'interpolation sur l'espace des couleurs, on ne risque pas d'ajouter de couleurs non présentes dans les données, mais on risque de manquer les hautes fréquences (zones à fortes variations de couleur et d'opacité). La post-classification reconstruit, quant à elle, l'ensemble des valeurs scalaires  $s$ , et donc l'ensemble des couleurs. La fonction de transfert est appliquée après interpolation, donc au niveau du champ de données. La visualisation qui en découle permet de rendre tous les détails de la fonction de transfert. Cependant cette approche est moins performante que la précédente vu l'application de la fonction de transfert à toutes les valeurs issues de l'interpolation linéaire inter-échantillons. [5].

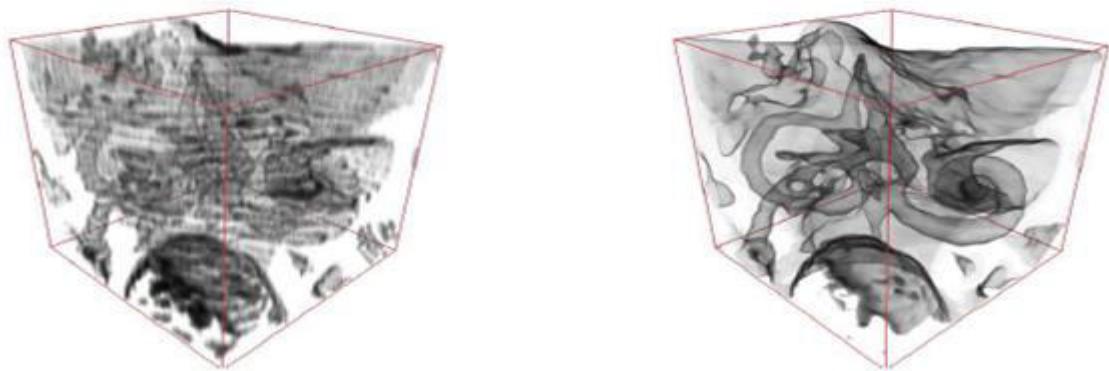
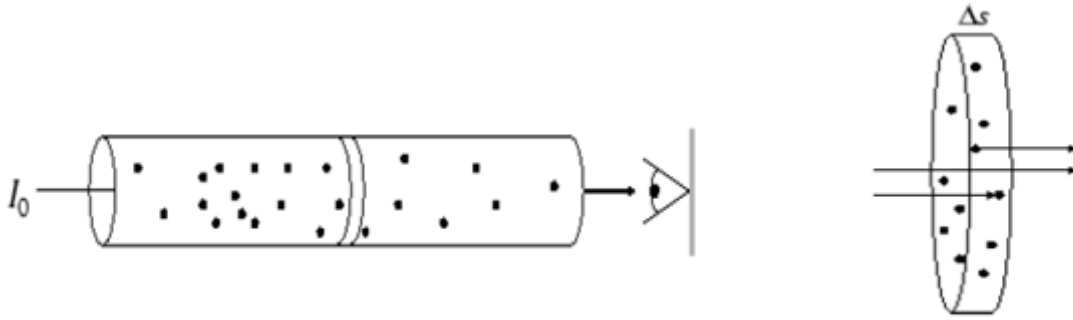


Figure 1.11 Comparaison de qualité d'images entre la pré-classification (Gauche) et la post-classification (Droite)

## 7.Équation de rendu volumique

Nous avons vu précédemment la manière dont on attribue des propriétés optiques aux valeurs scalaires, il nous faut maintenant évoquer la visualisation des données elles-mêmes. Pour cela, on va considérer le champ scalaire à trois dimensions comme un milieu participatif restreint.

Un milieu participatif ou milieu semi-transparent est un milieu qui influe sur le transport de la lumière en son sein. L'atmosphère, les liquides ou encore la fumée sont des exemples d'un tel milieu. A la base, un milieu participatif peut influencer sur la lumière qui le traverse de plusieurs manières: en l'absorbant, l'émettant ou la diffusant. Cependant, on ne considère ici que l'absorption et l'émission du milieu participatif, afin de simplifier les calculs. La perception de ce milieu par un observateur est alors formée par l'ensemble des rayons de lumière qui vont le traverser et atteindre l'observateur (**figure 1.12**).



a) Rayon traversant un milieu participatif

b) Tranche de données traversée

Figure 1.12 Schéma d'un rayon de lumière traversant un corps participatif [8]

L'expression décrivant la progression de l'intensité lumineuse portée par un rayon est appelée l'intégrale de rendu volumique. Pour l'obtenir, on définit tout d'abord la variation d'intensité lumineuse le long d'un rayon :

$$dI = -I_{absorbée} + I_{émise} \quad \text{Équation 1}$$

De là, on peut en déduire une équation différentielle, qui développée s'écrit alors sous la forme suivante (l'obtention et le développement est décrit par Max [9])

$$I(D) = I_0 \cdot e^{-\int_0^D \tau(t) dt} + \int_0^D c(s) \cdot \tau(s) \cdot e^{-\int_s^D \tau(t) dt} ds \quad \text{Équation 2}$$

Avec  $\tau(s)$  et  $c(s)$ , respectivement l'atténuation et la couleur émise au point  $s$ . La quantité  $e^{-\int (t) dt}$  décrit la transparence du milieu entre 0 et  $D$ . Le premier terme décrit l'intensité initiale (la lumière s'atténue exponentiellement en fonction de la distance  $D$  parcourue). Cette partie modélise donc l'absorption de la lumière (ou plutôt la lumière non absorbée). Le second terme décrit, quant à lui, l'ensemble de la lumière émise en chaque point sur le rayon de lumière en tenant compte du facteur d'atténuation, autrement dit la composition de l'ensemble des couleurs associées à chaque scalaire  $s$  sur le chemin du rayon. La résolution de l'équation précédente ne peut se faire analytiquement dans la majorité des cas. On recourt donc à une approximation numérique de cette intégrale la plus simple étant l'utilisation de la somme de Riemann, qui consiste à passer de :

$$\int_0^D h(x) dx \approx \sum_{i=0}^n h(x_i) \Delta x \quad \text{Équation 3}$$

En découpant l'intervalle de 0 à D en n intervalles et en sommant l'aire des rectangles formés, comme illustré avec la figure suivante:

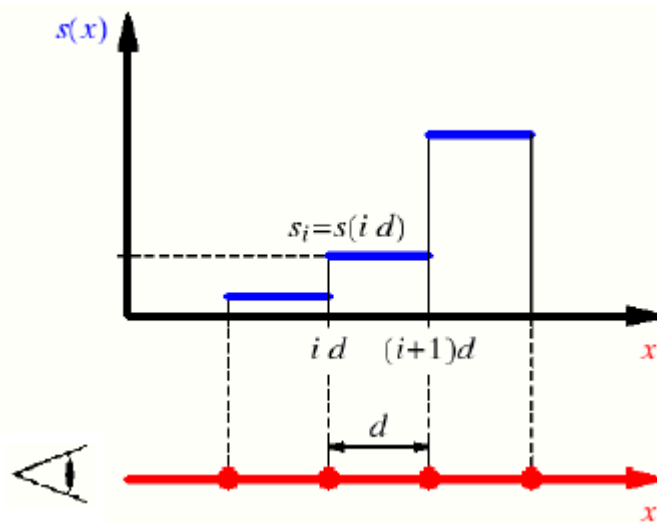


Figure 1.13 Approximation de l'intégrale par la somme de Riemann [10]

Après transformation de l'intégrale de rendu, on obtient l'expression suivante :

$$I \approx \sum_{s=0}^{n-1} c(s) \cdot \tau(s) \cdot \prod_{k=0}^{s-1} (1 - \tau(k)) \quad \text{Équation 4}$$

## 8. Les techniques de Raycasting

Il existe deux techniques principale de Raycasting : Maximum Intensity Projection et par accumulation (composition):

### 8.1. Maximum Intensity Projection

Dans l'algorithme MIP [11], les voxels sélectionnés sont ceux qui possèdent l'atténuation maximale le long des rayons projetées à travers l'ensemble de données volumiques. Le sous ensemble de ces voxels du volume est ensuite incorporé dans une image bidimensionnelle.

MIP est un outil de rendu largement utilisée et s'est révélée particulièrement utile pour l'évaluation et l'affichage dans l'angiographie CT [12]. MIP a été montré pour être plus précis que le rendu de surface pour évaluer le système vasculaire.

La figure suivante représente une angiographie par résonance magnétique visualisée par MIP :

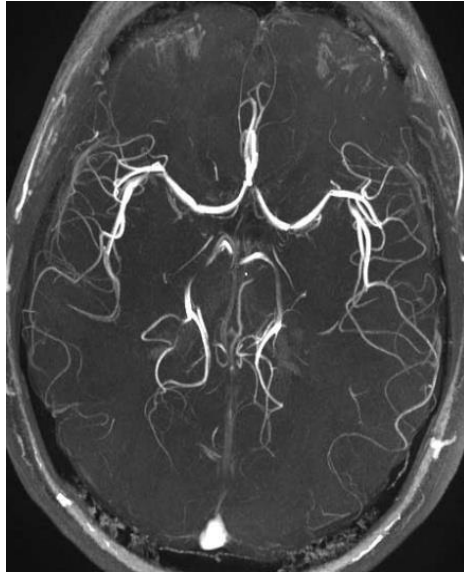


Figure 1.14 MIP d'une angiographie par résonance

## 8.2. Par accumulation (composition)

Le Raycasting par composition sert à accumuler les couleurs et les opacités rencontrés le long du rayon. Les valeurs de gris de l'ensemble de données sont affectées de différentes propriétés optiques (intensité de couleur) à travers la fonction de transfert pour obtenir l'image final qui peut être ensuite obtenue en évaluant l'intégrale de rendu en fonction de sortie de la fonction de transfert. Il existe deux types différents de la composition selon est le sens d'accumulation : d'avant en arrière et d'arrière en avant.

### 8.2.1. Composition d'avant en arrière

La figure suivante explique la composition d'avant en arrière:

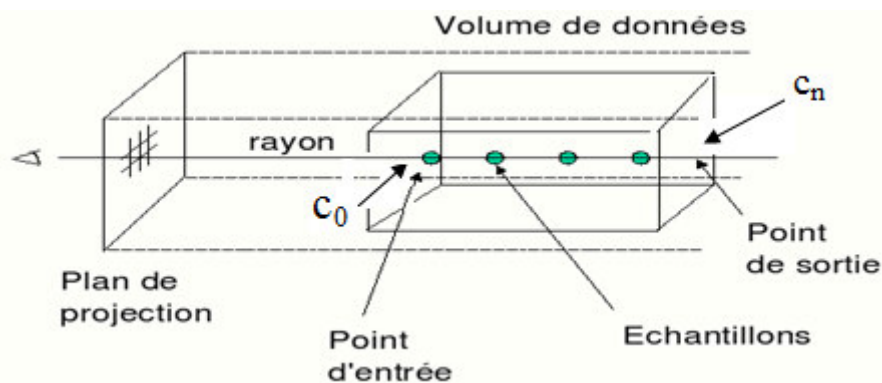


Figure 1.15 Composition d'avant en arrière

Ce qui nous donne la formule de composition suivante (avec  $I = C_n$ ) :

$$C_{i \text{ out}} = C_{i-1 \text{ out}} + (1 - \alpha_{i-1}) \cdot C_{i \text{ in}}$$

$$\alpha_{i \text{ out}} = \alpha_{i-1 \text{ out}} + (1 - \alpha_{i-1 \text{ out}}) \cdot \alpha_{i \text{ in}}$$

Avec  $c_0$  et  $\alpha_0$ , valeurs initiales respectivement égales à (0, 0, 0) et 0.

On doit dans cette méthode disposer d'un tampon supplémentaire pour l'accumulation des opacités  $\alpha_i$ . L'avantage majeur de cette approche est le fait de pouvoir arrêter de traiter des échantillons une fois que l'opacité a atteint 1 lors du parcours. En effet, si l'opacité est Maximale, le facteur  $1 - \alpha_{i-1}$  vaudra toujours 0 alors les échantillons suivant de  $i-1$  ne contribueront pas à la couleur finale. Ainsi si après avoir parcouru  $j$  échantillons, l'opacité cumulée a atteint le maximum, on peut alors éviter de traiter les  $n - j$  échantillons restants.

### 8.2.2. Composition d'arrière en avant

La seconde technique de composition prend le problème dans l'autre sens: on combine les échantillons de  $n - 1$  à 0, comme schématisé avec la "figure 1.16".

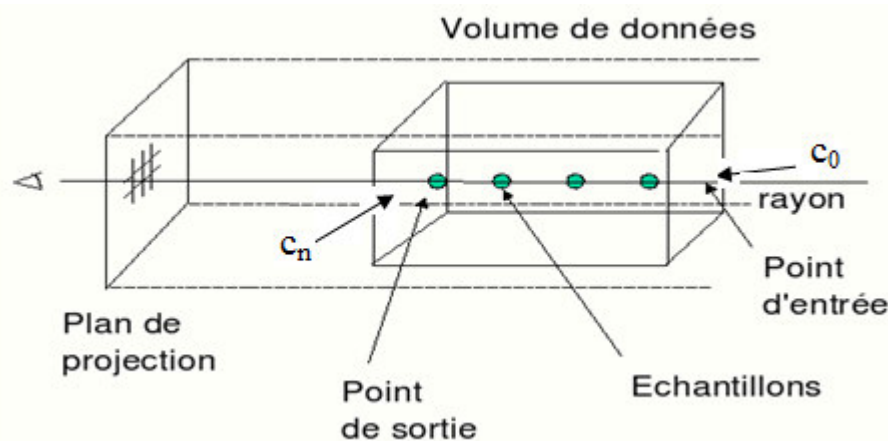


Figure 1.16 Composition d'arrière en avant

La formule de composition diffère alors de la composition d'avant en arrière (avec  $I = C_0$ ) :

$$C_{i \text{ in}} = C_{i \text{ in}} + (1 - \alpha_i) \cdot C_{i+1 \text{ out}}$$

Avec  $C_n$ , valeur initiale égale à (0, 0, 0).

Cette technique présente un inconvénient par rapport à l'autre méthode de composition: on doit traiter l'ensemble des  $n$  échantillons pour obtenir un résultat. En effet, du fait que l'on parcourt les échantillons d'arrière en avant, on ne pourra savoir si l'échantillon le plus en avant est totalement opaque (Il annule par conséquent la contribution des échantillons précédents) uniquement si on l'a traité [5].

## **9. Conclusion**

Dans ce chapitre on a présenté l'état de l'art de notre projet qui consiste à explorer le rendu volumique en générale et le Raycasting en particulier. Le chapitre suivant discutera la conception et l'implémentation de notre projet.



## 1. Introduction

Dans ce chapitre on va présenter le raycasting adaptatif, une définition puis une quelques méthodes dans le contexte.

## 2. Raycasting adaptatif

### 2.1 Définition

Le Raycasting adaptatif est une technique qui est fait pour améliorer le rendu par la réduction du nombre de points d'échantillonnage pour rendre de grands ensembles de données où chaque évaluation nécessite une décomposition coûteuse (diminuer l'échantillonnage dans les régions homogènes). Ainsi, on peut augmenter la fréquence d'échantillonnage dans les régions d'image où les variations des couleurs dans l'espace sont élevées.

### 3.Échantillonnage adaptatif:

Cette méthode a été proposé pour être utiliser dans les cas des isosurfaces. Afin de trouver la position d'intersection pour chaque rayon, la fonction scalaire est reconstruite à des positions d'échantillonnage discrètes  $x_i(x, y)c + t_i d(x, y)$  pour augmenter les valeurs de  $t_i$  dans  $[t_{start}, t_{exit}]$ . Notez qu'en général l'intersection exacte se situe quelque part entre deux échantillons successifs. En raison de cet échantillonnage discret, il est possible qu'une intersection soit totalement manquée lorsque le segment entre deux échantillons successifs croise deux fois l'isosurface.

Ceci est principalement un problème pour les rayons proches de la silhouette. Intersections garantie même pour des feuilles minces sont possibles si la longueur de gradient est limitée par une valeur  $L$  [15]. (**Voir Équation 1**). Notez que pour les champs de distance,  $L$  est égal à 1. Pour certains exemples de valeur  $f$ , il est connu que l'intersection à l'isovaleur  $\rho$  ne peut pas se produire pour un point plus proche de :

$$h = |f - \rho|/L. \quad \text{Équation 1}$$

Cependant,  $h$  peut devenir arbitrairement petit près de l'isosurface, ce qui donnerait un nombre infini d'échantillons pour des intersections garanties. L'échantillonnage adaptatif est pour améliorer la détection intersection. L'échantillonnage adaptatif utilisé pour améliorer la détection d'intersection. La position d'intersection réelle d'une intersection détectée est ensuite affinée en utilisant l'approche décrite dans la section 3.1.

On constate que les taux d'échantillonnage complètement et purement adaptatifs ne conviennent pas aux implémentations sur le matériel graphique. Ces architectures utilisent plusieurs pipelines où de petits bloc de pixels voisins sont converties en parallèle en utilisant la même mémoire cache de texture.

Avec un taux d'échantillonnage complètement adaptatif, les positions d'échantillonnage des pixels voisins divergent lors de l'exécution en parallèle, ce qui entraîne une utilisation implicite du cache. Par conséquent, cette méthode n'utilise que deux taux d'échantillonnage discrets différents. Le taux d'échantillonnage de base  $r_0$  est spécifié directement par l'utilisateur, où 1.0 correspond à un seul voxel. C'est le principal compromis entre la vitesse et l'épaisseur minimale de la feuille avec des intersections garanties. Afin d'améliorer la qualité des silhouettes, cette approche utilise un deuxième taux d'échantillonnage maximal  $r_1$  sous la forme d'un multiple constant de  $r_0$ :  $r_1 = nr_0$  où  $n = 8$ . Cependant, elle ne détecte pas explicitement de silhouettes, car cela serait trop coûteux. Par centre on augmente automatiquement le taux d'échantillonnage de  $r_0$  à  $r_1$  lorsque la valeur de l'échantillon actuel est plus proche de l'isovaleur  $\rho$  d'un petit seuil  $\delta$ .

Dans l'implémentation,  $\delta$  est défini par l'utilisateur comme paramètre de qualité, ce qui est particulièrement facile pour les utilisateurs. Champs de distance où la magnitude du gradient est 1.0 partout. Dans ce cas, une constante  $\delta$  peut être utilisé pour tous les jeux de données, alors que pour les scanners, elle doit être définie en fonction des données.



Figure 2.2. L'image de gauche illustre un petit détail du modèle de dragon asiatique avec un taux d'échantillonnage de 0,5. À droite, l'échantillonnage adaptatif augmente le taux d'échantillonnage à 4,0 proches de l'iso-surfaces. *N*

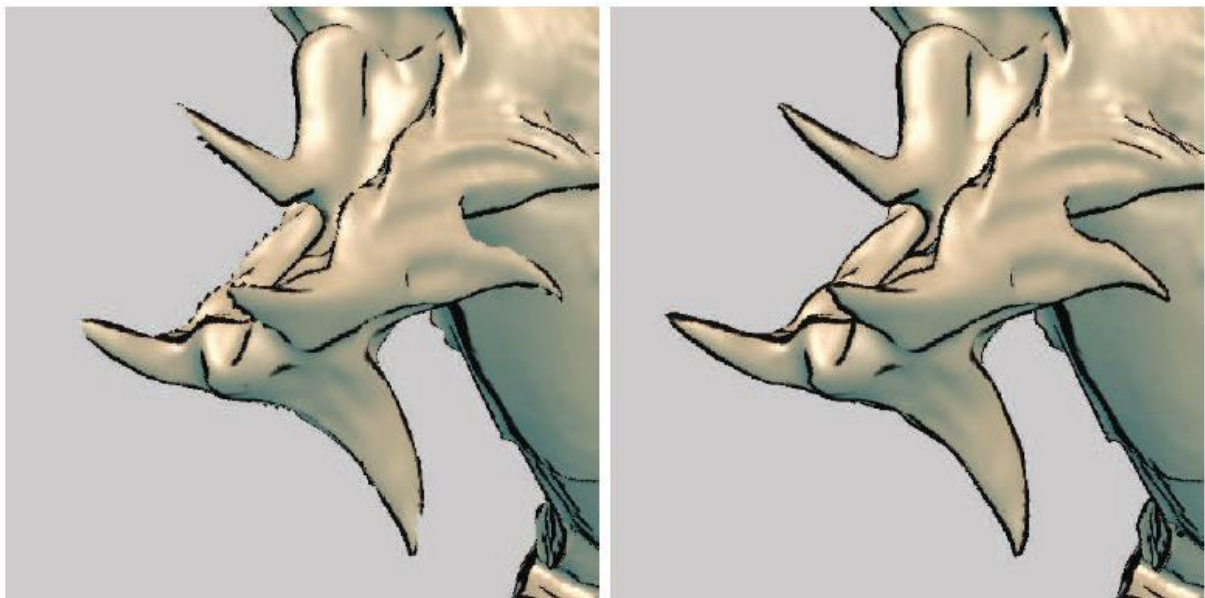


Figure 2.1. L'activation du raffinement de l'intersection isosurface entraîne une amélioration considérable de la qualité de l'image sans impact notable sur les performances.

### 3.1. Raffinement des intersections

Une fois qu'un segment de rayon contenant une intersection a été détecté, l'étape suivante détermine une position d'intersection précise à l'aide d'une procédure de bisection itérative. Lors d'une itération, on calcule d'abord une position d'intersection approximative en

supposant un champ linéaire dans le segment. Étant donné les valeurs d'échantillon  $f$  aux positions  $x$  pour les extrémités proches et éloignées du segment, la nouvelle position d'échantillon est

$$x_{new} = (x_{far} - x_{near}) + \frac{\rho - f_{near}}{f_{far} - f_{near}}$$

Ensuite, la valeur  $f_{new}$  est extraite à ce stade et comparée à l'isovaleur  $\rho$ . En fonction du résultat, on mis à jour le segment de rayon avec le sous-segment avant ou arrière. Si le nouveau point se trouve devant l'isosurface (par exemple,  $f_{new} > \rho$ ), défini  $x_{near}$  sur  $x_{new}$ , sinon défini  $x_{far}$  sur  $x_{new}$  et répétez. Souvent, un nombre fixe d'étapes, par exemple quatre étapes, est suffisant pour obtenir des positions d'intersection de haute qualité.

## 4- Multirésolution basée Niveau de détail (LOD) et l'échantillonnage adaptatif

### 4.1. Représentation multirésolution basée sur octree :

Le schéma de stockage linéaire décrit ci-dessus a une propriété de localité de données médiocre. La recherche de voxels voisins est fréquente et ce n'est que le long de l'axe des  $x$  que cela se traduit par un accès aux emplacements de mémoire voisins. L'impact des erreurs de cache dans le rendu et le traitement est important et fait souvent échouer un schéma s'il n'est pas correctement géré. Le blocage du volume est donc généralement efficace et améliore considérablement le taux d'accès au cache.

La taille d'un bloc est généralement calculée à partir de la taille des caches de niveaux 1 et 2. Constate qu'une taille de bloc de 32,  $B = (32, 32, 32)$ , est la plus efficace pour leur raycaster basé sur des blocs. Parker et al. [15] utilise une taille de bloc plus petite,  $B = (4, 4, 4)$  pour un rendu d'isosurface parallèle. Cette taille de bloc correspond à la taille de la ligne de cache L1 sur les super-ordinateurs SGI. Dans de nombreuses publications, il est indiqué que bloquer par 16 ou 32 est la taille optimale pour de nombreuses tâches de traitement liées aux blocs.

L'adressage de blocs et d'échantillons dans des blocs est simple, mais l'introduction d'une structure de placage de blocs permet un placement arbitraire de blocs dans la mémoire, les blocs inutilisés étant ignorés, ce qui permet d'économiser de l'espace. L'introduction du blocage entraîne un niveau supplémentaire de complexité pour la gestion des frontières de

blocs, en particulier dans les cas où un échantillon est demandé dans un bloc voisin qui a été ignoré. Deux stratégies peuvent être explorées pour y remédier.

La première nécessite l'accès des blocs voisins. Par exemple, proposez un schéma basé sur des recherches de table pour les échantillons voisins qui évite les branches conditionnelles dans le code. La deuxième stratégie est basée sur des blocs autonomes et nécessite la réplication d'échantillons voisins. Le temps système nécessaire à la réplication des échantillons est inférieur à 20% pour les blocs de taille supérieure ou égale à 16.

Des conversions de données de base supplémentaires peuvent également être appliquées, telles que le remappage du rang de valeurs et la conversion en entiers à 8 ou 16 bits, c'est-à-dire des types de données mappés directement aux types de GPU natifs. La subdivision en blocs améliore la localisation en mémoire pour le rendu et le traitement logiciels.

Ignorer les blocs vides a généralement un effet important sur la taille des données et les performances de rendu. Le concept de bloc vide doit cependant être clarifié et défini, ce qui fait partie intégrante de la section suivante.

## **4.2. Blocs et structures d'accélération**

Afin de révéler toutes les entités dans un volume, il est évident que certains échantillons doivent être rendus transparents et d'autres échantillons rendus semi-transparentes ou opaques. Ceci est réalisé grâce à l'utilisation d'une fonction de transfert (FT). Pour un schéma de blocage, tel que décrit ci-dessus, la signification d'un bloc vide est un bloc dont tous les voxels sont classés comme complètement transparents. Naturellement, un tel bloc pourrait être supprimé dans le processus de rendu et améliorer ainsi les performances. L'objectif est de réduire la quantité de données dans le pipeline, il est essentiel que les blocs vides puissent être prédits sans accès à tous les échantillons d'un bloc. Les métadonnées pour telles prédictions sont collectées pendant le prétraitement, et de préférence sans connaître les paramètres de FT spécifiques.

La FT définit généralement une ou plusieurs régions de la plage scalaire comme non transparentes et pour le rendu des iso-surfaces, des « narrow peaks » sont définis ou des rendus spéciaux par iso-surfaces sont utilisés. Il est donc naturel que les idées des méthodes d'accélération d'extraction par extraction iso-superficielle aient été appliquées. Le but de ces schémas est de minimiser le traitement afin que seules les cellules intersectées l'iso-surface soient prises en compte. Wilhelms & Gelder créent un arbre de valeurs min / max. L'arbre est

créé de bas en haut et commence par les cellules, des cubes de 8 voxels. Livnat et al. [15] Etendent cette approche et introduisent le span-space. Pour le rendu iso-surfacique, une feuille dans l'arborescence est incluse si la iso-valeur est comprise dans la plage couverte par les valeurs minimale et maximale de la cellule. Parker et al. [15] utilisent un arbre limité à deux niveaux et trouvent que cela suffit dans la mise en œuvre logicielle d'un diffuseur de rayons iso-surface. Pour les paramètres FT arbitraires, le schéma min / max est généralement trop prudent et peut classer les blocs vides comme non vides. On utilise les tableaux de surfaces sommées pour déterminer le contenu des blocs en prenant la différence entre les entrées de tableau pour les valeurs de bloc minimale et maximale.

L'inconvénient de l'approche min/max [15] est qu'ils utilisent plutôt un vecteur binaire pour identifier le contenu du bloc. Le rang La plage scalaire est quantifié en 32 régions uniformes et le vecteur de bits indique la présence d'échantillons dans la plage correspondante. Mais ils utilisent un vecteur plus grand, correspondant à la taille de leur table FT (256 entrées).

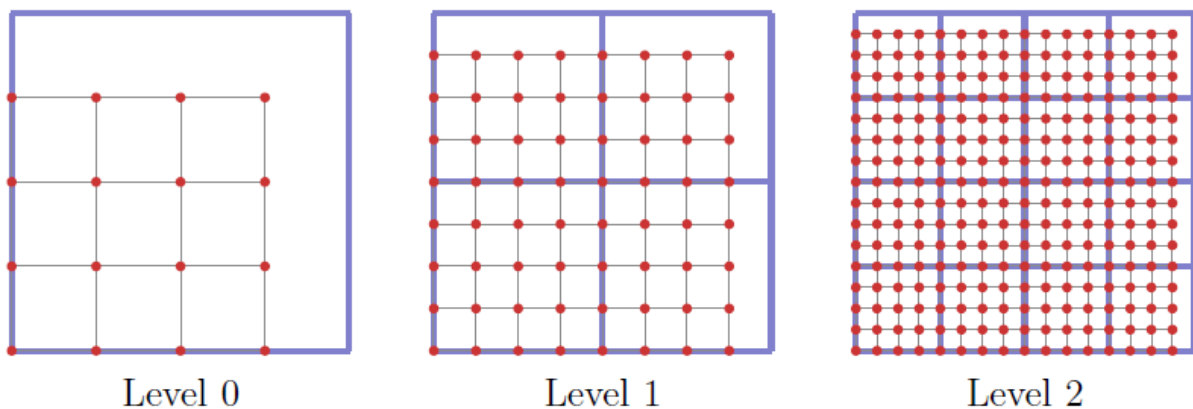


Figure 2.3 : Division en blocs hiérarchique avec sous-échantillonnage.

### 4.3. Représentations multirésolution hiérarchiques

Ignorer simplement des blocs vides risque de ne pas réduire suffisamment la taille du volume. La taille totale des blocs non vides restants peut toujours être supérieure à la taille de la mémoire disponible. Une stratégie consiste ensuite à appliquer des techniques faisant varier la résolution dans différentes parties du volume, afin que différents blocs du volume ayant des résolutions différentes. Cette approche de niveau de détail (LOD) permet une adaptation plus gracieuse aux ressources de mémoire et de traitements limités.

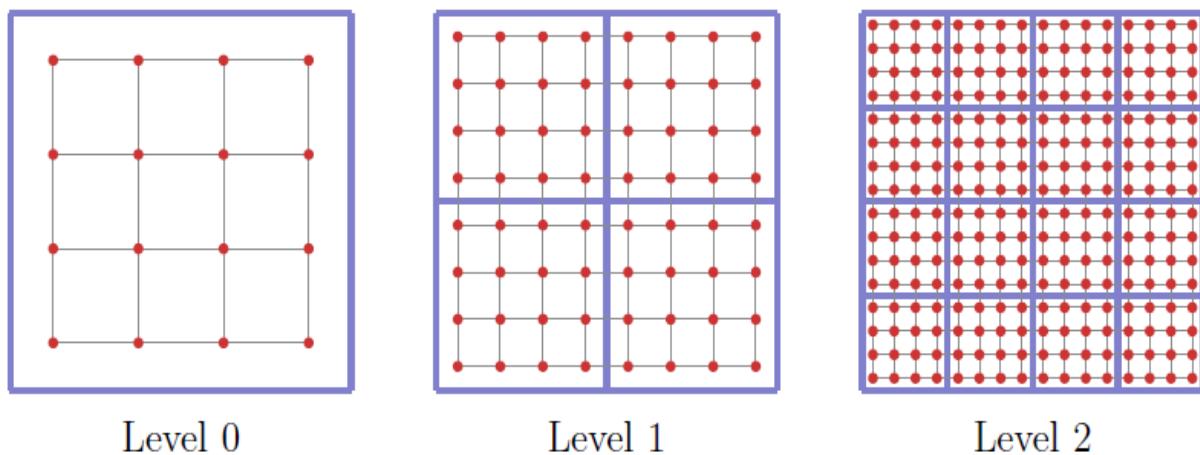
Le schéma le plus courant consiste à créer une représentation hiérarchique du volume par sous-échantillonnage récursif du volume d'origine. Étant donné que chaque niveau de résolution inférieur correspond à 1/8 de la taille du précédent, la quantité de mémoire



supplémentaire requise pour cette pyramide est inférieure à 14,3%. Les hiérarchies créées peuvent différer selon le schéma de sous-échantillonnage sélectionné. La **figure 2.3** illustre trois niveaux d'une hiérarchie créée à l'aide d'un sous-échantillonnage, chaque deuxième échantillon étant supprimé.

Les positions des valeurs sous-échantillonnées nécessitent toutefois une certaine attention. Le positionnement indiqué à la **figure 2.3** incline le domaine représenté. Un placement plus approprié d'une valeur sous-échantillonnée se trouve au centre des valeurs de résolution plus élevées qu'il représente. Ce placement est illustré à la **figure 2.4** et s'appuie également sur un sous-échantillonnage moyen.

Pour pouvoir sélectionner différents niveaux de résolution dans différentes parties du blocage de volume, il convient également aux représentations hiérarchiques. La taille de bloc, en termes de nombre d'échantillons, est généralement maintenue égale à chaque niveau de résolution et les grilles de bloc sont indiquées par de larges lignes bleues dans les **figure 2.3** et **figure 2.4**. Les blocs à basse résolution couvrent des étendues spatiales de plus en plus grandes du volume. Ces hiérarchies multirésolution fournissent ainsi des structures de données de support pour la sélection de LOD.



*Figure 2.4. Division en blocs hiérarchique avec sous-échantillonnage moyen.*

## 5.Échantillonnage de volumes reconstruits par la multirésolution

La structure multirésolution plate fournit un schéma d'adressage uniforme pour l'accès aux blocs et aux échantillons. Le rang de volume est défini par le nombre de blocs,  $N_x$ ,  $N_y$ ,  $N_z$ , le long de chaque dimension. L'indice de bloc  $p$ , est alors facilement récupéré en tant que partie entière d'une position,  $p$ , dans le volume.

Le reste de la position définit alors la coordonnée locale intra-bloc,  $p' = \text{frac}(p)$ . La mappe d'index de blocs, contenant la taille de bloc E et l'emplacement Q du bloc dans le volume condensé, est ensuite utilisée pour calculer la coordonnée à prendre par l'échantillon. Le rang de coordonnées pour l'adressage de blocs plats est défini en unités de blocs. Prendre la partie entière d'une coordonnée donne alors l'indice de bloc et le reste donne la coordonnée locale intra-bloc.

L'index de bloc est utilisé pour rechercher l'échelle et l'emplacement d'un bloc dans le volume condensé. Étant donné que les blocs du volume condensé sont rarement voisins du domaine spatial, un échantillonnage particulier doit être effectué lors de l'échantillonnage d'un bloc. En outre, l'échelle d'un bloc est arbitraire, ce qui a également des implications pour l'échantillonnage par bloc. Une illustration d'un voisinage de quatre blocs est illustrée à la **figure 2.5**.

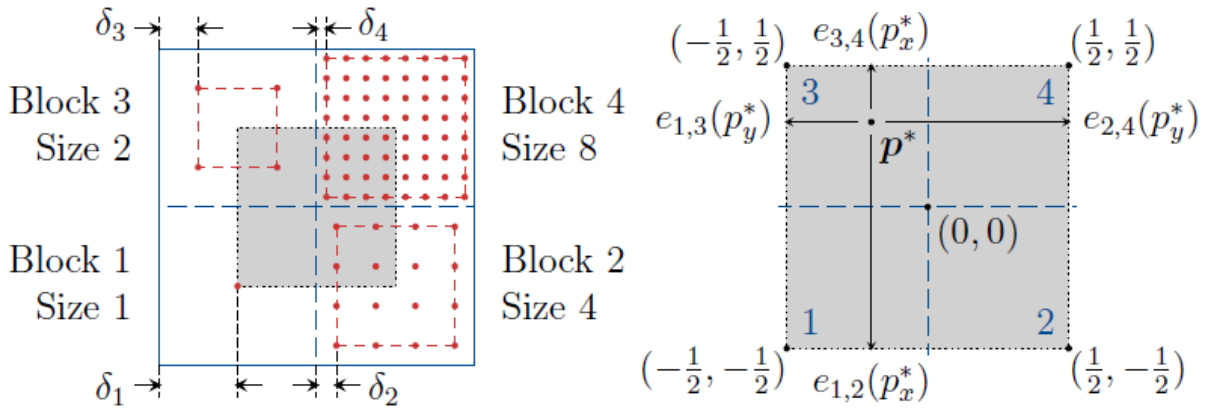


Figure 2.5. Illustration des voisins de bloc en 2D

### 5.1.Échantillonnage de bloc le plus proche

La première approche de l'échantillonnage par blocs consiste à restreindre l'échantillonnage pour accéder uniquement aux données du bloc actuel, exprimées de manière appropriée en termes d'opérations sur la coordonnée intra-bloc,  $P'$ . Le domaine de coordonnées valide pour les échantillons intra-blocs est indiqué par des carrés de lignes rouges en pointillés sur la **figure 2.5**.

L'insert des frontières spatiales des blocs est indiqué par  $\delta_i$  pour chaque bloc  $i$ . Le lieu d'échantillonnage restreint  $p_c$  est alors défini comme suit:

$$p_c = c_{1-\delta}^{\delta}(p')$$

Où  $C_{1-\delta}^{\delta}$  fixe la valeur ou les valeurs d'élément pour les vecteurs de  $x$  à l'intervalle  $[\alpha, \beta]$ , Cette forme d'échantillonnage a été renommée Échantillonnage du bloc le plus proche (NB) [15]. Il est évident que des artefacts de bloc vont se former, mais ils ne sont pas toujours visibles et il est indiqué ci-dessous qu'ils sont les plus apparents pour les réglages de surface mince, iso-superficiels, comme FT.

## 5.2. Échantillonnage d'interpolation interbloc

Pour remédier à ces artefacts de blocs, une technique d'interpolation d'interblocage a été développée [15]. Les techniques précédentes reposent sur la réplique d'échantillon et le remplissage entre les blocs. Cependant, la réplique limite la réduction des données dans le pipeline et peut également fausser les échantillons lorsqu'un bloc a une résolution supérieure à celle du voisin, afin de réduire les discontinuités d'interpolation. Plus précisément pour les blocs inférieurs de résolution, et implicitement pour la réduction des données plus la surcharge de données devient de plus en plus importante.

Interbloc Interpolation (II) supprime la nécessité pour la réplique de l'échantillon est un système d'interpolation directe entre les blocs de résolution arbitraire, y compris les non-puissances de-deux tailles. Le principe de l'échantillonnage II consiste à prélever un échantillon de chacun des blocs voisins les plus proches immédiats et à calculer une valeur d'échantillon à l'aide d'une somme pondérée normalisée. Le domaine d'interpolation entre blocs dans un voisinage est indiqué par la zone ombrée située entre les centres des blocs dans la figure 2.5. La coordonnée locale interbloc,  $p^* = \text{frac}(p + 0.5) - 0.5$ , a son origine à l'intersection des blocs adjacents.

Le poids de bloc  $\omega_b$ , pour chacun des blocs, est calculé en utilisant des poids de bord individuels,  $e_{i,j}$ , pour les bords situés entre deux centres voisins du bloc faisant face, les blocs  $i$  et  $j$ , comme illustré en 2D à la **figure 2.5** par les bords en gris de la boîte en pointillés entre centres de bloc. La **figure 2.6** montre quatre schémas de pondération des arêtes différents, y compris le mode d'échantillonnage NB. Le schéma Distance maximale fournit l'interpolation la plus douce, mais également le plus sensible au mauvais choix du niveau de détail, car le noyau de filtre possède le support pondéré le plus large. Considérons, par exemple, le cas où une iso-surfaces coupe un bloc en dehors de sa limite d'échantillon. Si un bloc voisin est choisi à un niveau de détail faible, la surface se pliera vers le bloc à faible résolution. Par

conséquent, la distribution des valeurs de bloc utilisée pour la sélection du niveau de détail comprend une couche d'échantillons autour de chaque bloc.

La version 2D du schéma d'échantillonnage est ensuite définie succinctement par la somme normalisée suivante:

$$\varphi = \frac{\sum_{b=1}^4 \omega_b \varphi_b}{\sum_{b=1}^4 \omega_b}$$

Où  $\varphi_b$  est un échantillon NB du bloc  $b$  et les poids de bloc  $\omega_b$ , sont définis comme suit :

$$\omega_1 = (1 - e_{1,2}) \cdot (1 - e_{1,3})$$

$$\omega_2 = e_{1,2} \cdot (1 - e_{2,4})$$

$$\omega_3 = (1 - e_{3,4}) \cdot e_{1,3}$$

$$\omega_4 = e_{3,4} \cdot e_{2,4}$$

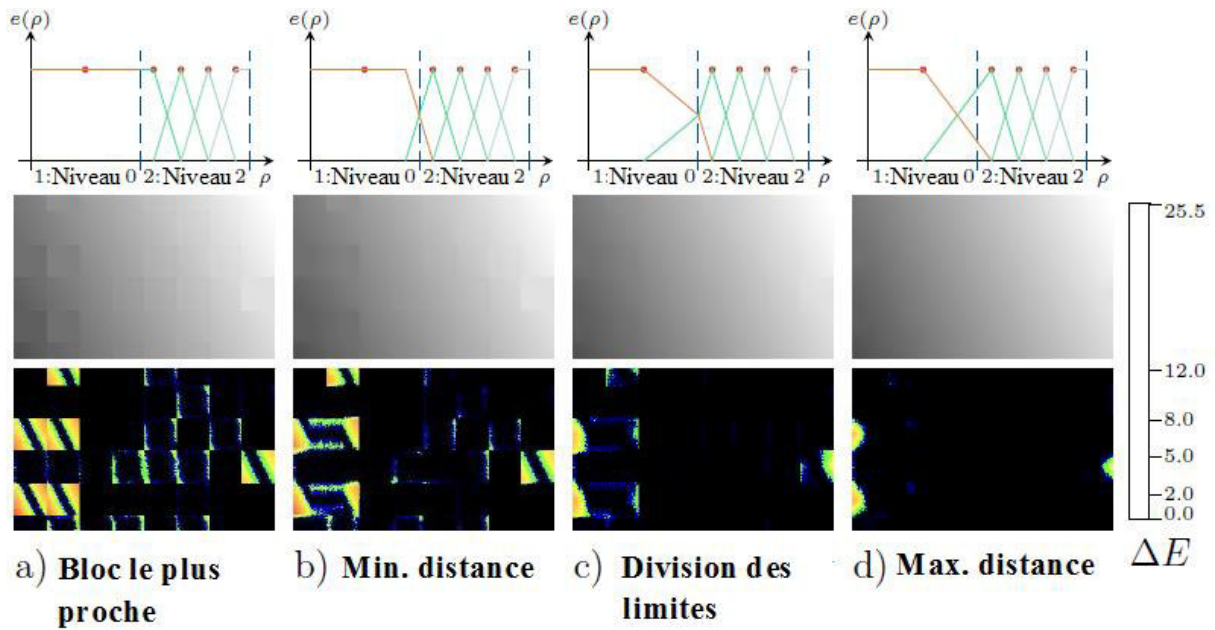


Figure 2.6. Comparaison des méthodes d'interpolation

La variante 3D correspondante et les définitions des fonctions de pondération des arêtes sont présentées au document. Toutes les fonctions de pondération des arêtes décrites dans ces dernières donnent des interpolations équivalentes à une interpolation trilinéaire lorsque les blocs voisins ont tous la même résolution. Il est également montré que la méthode constitue une fonction continue  $C^0$ .

### **5.3. Résultats interbloc d'interpolations**

La qualité de l'échantillonnage par interpolation interbloc est illustrée à la **figure 2.6**. Un « slightly rotated gradient » est utilisé pour évaluer les différentes fonctions des poids des arêtes et la rangée minimal indique l'erreur en pixels dans l'espace colorimétrique CIEL\*u\*v\*. [15]

La valeur maximale, avec FT définissant une iso-surface opaque, montre clairement des artefacts en bloc sans échantillonnage II. Le paramètre FT plus doux pour la rangée minimal indique des artefacts moins perceptibles.

Le schéma d'interpolation interbloc multirésolution a été initialement déployé dans un moteur de rendu basé sur le découpage en texture. L'augmentation des exigences de calcul pour l'échantillonnage II diminue les performances, environ 8-10. Il est ainsi possible de rendre chaque tranche en deux passes.

La première passe échantillonne le domaine intérieur et élimine le fragment pour les échantillons situés en dehors de l'intervalle de sélection des échantillons. Le second passage se remplit alors dans les échantillons en dehors des limites de l'échantillon en exploitant au début Z-termination. Le coût du shader d'échantillonnage II est plus cher et donc considérablement réduit l'impact sur les performances et réduit à un facteur de seulement 3 à 4, par rapport à « l'échantillonnage de bloc le plus proche » uniquement. Dans ce cas, il est important que les blocs à la résolution la plus basse soient entièrement échantillonnés avec un « l'échantillonnage de bloc le plus proche ».

## 6. Raycasting sur le GPU

Dans cette section, plusieurs techniques sont décrites qui améliorent de manière significative les performances de rendu pour la diffusion de rayons basée sur GPU, telles que présentées dans le document. Exploitant la flexibilité des GPU programmables, plusieurs techniques de classification avancées combinant plusieurs FT dans un raycasting, sont brièvement décrites. Tous les détails se trouvent dans [15].

### 6.1. Échantillonnage adaptatif dans l'espace objet

Comme on peut le voir dans les images de données volumiques, il existe souvent de grandes zones d'espace vide et, pour les volumes multirésolution, des blocs à une résolution inférieure à la résolution maximale. Il est évident que les performances de rendu peuvent être améliorées si ces pièces sont échantillonnées plus faiblement ou si elles sont totalement ignorées. C'est l'objectif de nombreuses recherches et plusieurs travaux ont été proposés, impliquant la cohérence spatiale, Ainsi que l'utilisation de structures de données d'accélération telles que les octrees. En général, ces approches ont été généralement des décisions binaires, une région soit ignorée, soit échantillonnée à pleine de densité.

Le pas adaptatif le long du rayon où les métadonnées de bloc [15], le niveau de résolution de chaque bloc, sont utilisées pour ajuster la densité des échantillons. La sélection du niveau de détail est en fait une structure de données d'accélération à un grain avec une adaptation plus continue, en particulier lorsqu'elle ne rend pas uniquement des surfaces isovaleur. L'amélioration théorique obtenue grâce à cette méthode représente un gain de performances proportionnelles à la racine cubique du facteur de réduction des données, car l'échantillonnage n'est réduit que dans une dimension le long des rayons.

Ensemble de données	GPU	TS	Plein	Limité	Naïve	Théorique
Femelle 11:1	ATI	5.3	5.1	14.1 (2.8)	19.8 (3.9)	2.2
Femelle 11:1	NV	3.6	2.3	6.0 (2.6)	9.7 (4.2)	2.2
Femelle 30:1	ATI	5.5	5.2	17.4 (3.3)	27.5 (5.3)	3.1
Femelle 30:1	NV	3.6	2.3	6.5 (2.8)	11.1 (4.8)	3.1
Crabe 9:1	ATI	3.4	2.9	6.2 (2.1)	8.7 (3.0)	2.1

Tableau 2.1. Performances de rendu d'échantillonnage adaptatif dans un espace objet



Dans une approche naïve, le pas à pas est dicté par la résolution du bloc en cours d'échantillonnage. Un échantillon proche de la limite du bloc peut alors potentiellement faire avancer le rayon profondément dans un bloc voisin ayant une résolution significativement plus élevée, ce qui donne une région sous-échantillonnée. La méthode est donc modifiée pour limiter le pas aux limites des blocs. Le raycaster a été développé pour réaliser l'échantillonnage par interpolation interbloc. Dans ce cas, le pas à pas est limité en fonction du nombre minimum d'étapes parmi les huit blocs voisins [15].

L'échantillonnage adaptatif en espace objet améliore considérablement les performances du raycasting basé sur un processeur graphique, tout en maintenant une qualité visuelle élevée. D'après le **tableau 2.1**, l'accélération dépasse légèrement l'accélération théorique attendue (dernière colonne). L'approche naïve par étapes montre des gains encore plus élevés, indiquant clairement qu'un sous-échantillonnage indésirable se produit.

L'application de l'échantillonnage II réduit les performances et des tentatives ont été faites pour contrer cet effet en exécutant de manière conditionnelle l'échantillonnage II dans le shader. En raison de la nature SIMD des unités de traitement de fragments, le traitement de nombreux fragments en parallèle rend ces efforts souvent contre-éfficaces en ce que les fragments ne se comportent souvent pas uniformément sur des carreaux suffisamment grands. Bien que cela pose un problème dans la génération actuelle de GPU, la granularité d'exécution est progressivement affinée.

## **7. Implémentation de l'Occlusion ambiante locale :**

Comme on peut le constater à partir des équations ci-dessus, le calcul de LAO est déterminé par le traitement de chaque voxel dans le volume. Une implémentation utilisant des volumes linéaires réguliers entièrement échantillonné est donc assez simple. Il est cependant inutile de calculer l'éclairage détaillé dans des espaces vides ou homogènes, telles que définies par FT. Les superflus d'échantillonnages dans ces régions peuvent être évités en employant une approche multirésolution.

Une telle approche peut supprimer des blocs vides et sélectionner des résolutions inférieures pour régions homogènes. Dans cette implémentation, la méthode multirésolution bloquante est utilisée [16], ce qui permet une réduction des données en utilisant le FT pour optimiser la qualité d'image résultante. Certaines structures de données et contraintes supplémentaires sont

nécessaires, décrit ci-dessous en détail, pour appliquer le programme LAO méthode dans le cadre multirésolution.

### **7.1. Traitement multirésolution**

Dans une étape de prétraitement, un volume linéaire régulier est divisé en petits blocs de  $16^3$  voxels, chaque bloc est codé avec tous les niveaux de détail, soit  $8^3$ ,  $4^3$ ,  $2^3$ ,  $1^3$  voxels. Ces blocs sont organisés en une représentation multirésolution permettant un chargement efficace de la sélection LOD arbitraire lors de l'exécution. La sélection de LOD est mise à jour lorsque le FT est changé ou le volume est éclaté. La résolution de chacun du bloc est choisie selon un processus d'optimisation basé sur le contenu des blocs dans le domaine FT, c'est-à-dire la distribution de donnée mappée via la FT, ce qui optimise la qualité d'image résultante étant donné l'utilisateur ou le système une limite de mémoire.

Le programme donnera la priorité aux blocs présentant une forte variation de la FT domaine, les blocs moyennes à faible variation sont donnés plus bas priorité, et les blocs vides peuvent être ignorés complètement. Dans cette mise en œuvre, les LOD minimum et maximum pour les blocs non vides peuvent être définis avec un LOD séparé pour les blocs vides. Les blocs vides peuvent également être ignorés. Les données de volume dans la sélection LOD sont regroupées dans une texture 3D unique, la traduction de coordonnées de volume  $V$  aux coordonnées de texture compactée  $P$  est facilitée par une texture d'index de mappage vers l'avant. Cette texture a la même taille que le nombre de blocs en trois dimensions et contient le décalage 3D et la taille de chaque bloc. Afin de faire l'inverse de traduction, l'inversion la cartographie indice de texture est également nécessaire. Depuis un seul voxel dans la texture de volume condensé peut couvrir un bloc entier, cette texture d'index de mappage inverse devrait être de la même taille que le volume condensé. Pratiquement, il est souvent suffisant pour restreindre le minimum LOD pour non vide blocs 43 voxels, et donc, permettre l'inverse indice de texture à 64 fois plus petits.

### **7.2. Pipeline LAO**

Une texture 3D est créée pour contenir la couleur émissive et la lumière ambiante de chaque voxel. Le volume est alors traité en mappant toutes les tranches une par une sous forme de rendu cible d'un objet framebuffer. À chaque passage, une ou plusieurs directions de rayons peuvent être traitées et déclenchées par le rendu d'un quadruplet sur la mémoire tampon, où chaque pixel est mappé sur un voxel dans la texture 3D mappée. La texture d'éclairage 3D a

les mêmes dimensions que la texture du volume emballé et la même présentation multirésolution est utilisée. Étant donné que le traitement est piloté par la structure multirésolution, la réduction de données obtenue par la sélection de niveau de détail LOA implique donc directement un gain de performance dans le calcul LAO.

L'estimation du LAO volumétrique est effectuée de manière incrémentielle, en mettant à jour une direction de rayon par trame et en enregistrant le résultat accumulé dans une texture lumière ambiante en utilisant l'opération de mélange OpenGL. La mise à jour progressive de la texture d'intensité ambiante, est reformulée comme suit :

$$A^k = \frac{1}{k} \cdot A_k + \left(1 - \frac{1}{k}\right) \cdot A^{k-1}$$

Où  $A^k$  est l'intensité accumulée après les  $k$  directions des rayons, avec  $A^0 = 0$ . L'opération de fusion ajoute une fraction de l'intensité  $A^k$  de la direction du rayon  $k$ .

Le nombre de rayon de directions  $k$  à échantillonner peuvent être configuré dynamiquement et les directions sont stockées dans une texture avec le poids directionnel  $w_k$ . Les directions des rayons sont créées en subdivisant un tétraèdre, un icosaèdre ou un octaèdre à différents niveaux, en fonction du nombre de rayons souhaité [16]. Ces formes génèrent des distributions uniformes pour les directions des rayons. Une fois que toutes les directions sont traitées, la lumière ambiante finale le volume d'intensité est utilisée dans la passe de rendu.

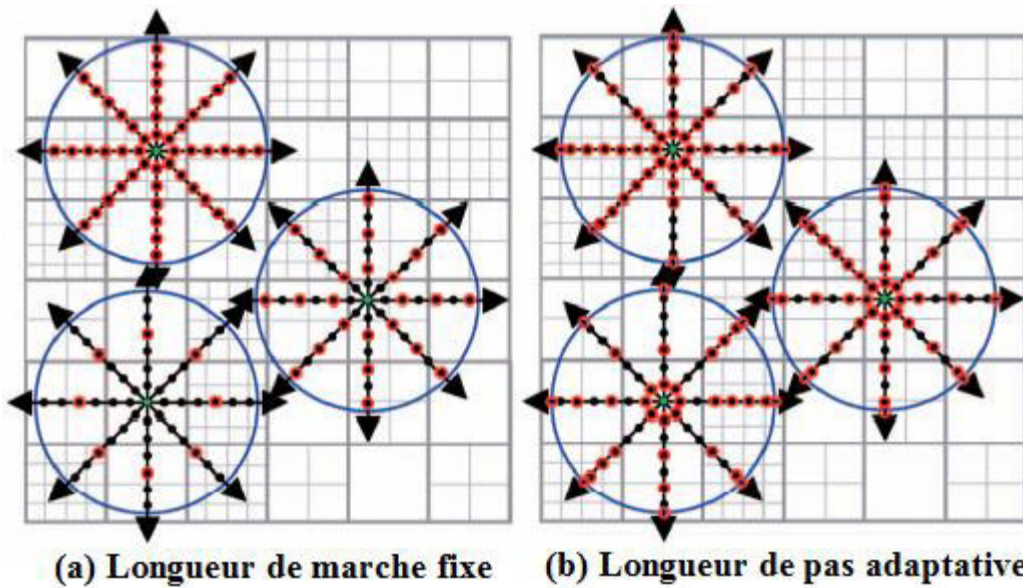


Figure 2.7. comparaison entre l'échantillonnage adaptatif et l'échantillonnage fixé

L'utilisation de l'emplacement du fragment dans le framebuffer plus le z-décalage de la tranche actuelle, l'inverser la carte  $P \rightarrow V$  est utilisé pour déterminer le voxel emplacement en coordonnées de volume. L'intégrale du rayon est alors calculée pour une ou plusieurs directions. Une direction par passe est couramment utilisée pour fournir un plus haut degré de l'interactivité. Dans l'intégration, chaque échantillon nécessite une texture indirection des coordonnées de volume a emballé les coordonnées de texture auxquelles se termine le mappage texture d'index  $V \rightarrow P$  est utilisé.

Le rendu de l'image en volume s'effectue avec Raycasting utilisant l'échantillonnage adaptatif, [16] cette technique utilise l'échelle de chaque bloc pour dynamiquement adapter le taux d'échantillonnage par bloc, et donc permet d'accélérer l'ordre de la racine cubique des données de taux de réduction, le schéma est donc capable de sauter l'espace vide à l'avant et à l'arrière du volume, ainsi que tout espace vide dans le volume. Pour fournir un champ scalaire continu à partir du volume multirésolution, La technique d'interpolation interbloc a été utilisée [16]. Cette approche ne nécessite aucune réplification de voxel à partir de blocs voisins et prend également en charge les différences de résolution arbitraires. Ce schéma d'interpolation est appliqué lorsque l'utilisateur cesse d'interagir avec le volume. Dans des nombreux cas, les artefacts de discontinuité sont dépendent sur le FT actuel. L'interpolation entre blocs n'est cependant pas prise en compte dans les calculs d'LAO, car il faudrait trop de textures. Seuls des artefacts mineurs ont été observé en raison de cette limitation.

### 7.3.Échantillonnage d'occlusion locale

La performance des calculs de LAO peut être améliorée en ajustant la densité d'échantillonnage le long des rayons pour les voxels dans les blocs de résolution inférieure. Il y a différentes manières d'ajuster la densité d'échantillonnage et l'approche la plus simple consiste à ajuster le taux d'échantillonnage par rapport à la densité d'échantillonnage des données de volume sous-jacentes. Comme illustrée dans la **figure 2.7 (a)**.

Pour un rayon provenant d'un bloc d'échelle  $k$ , la longueur du pas est ajustée  $k_{max}/k$  pour tous les échantillons le long de ce rayon. La résolution maximale  $k_{max}$  donne l'échantillonnage le plus élevé densité, étant donné la densité d'échantillonnage de base  $\Delta_B$ .

Cela peut conduire à des artefacts car les blocs de basse résolution peuvent manquer des caractéristiques d'occlusion dans les blocs de voisins résolution due au sous-échantillonnage alors qu'un voisin Le bloc de basse résolution est fortement sur-échantillonné. En utilisant une densité d'échantillonnage adaptative basée sur le bloc la résolution  $k_i$  à l'emplacement de l'échantillon  $i$  le long du rayon, ces artefacts sont réduits. Cela garantit que les fonctionnalités ne sont pas manquées, étant donné qu'une densité d'échantillonnage de base suffisante est utilisée. **figure 2.7 (b)** illustre cette stratégie d'échantillonnage adaptatif.

Une mise en garde est le franchissement des limites de blocs dans lequel le schéma doit s'assurer que la densité est correctement ajustée lorsque entrer dans un nouveau bloc. En marchant le long du rayon, la prochaine position est définie sur le premier emplacement discret, en fonction de la densité d'échantillonnage de base, comprise dans le bloc suivant [16]. Cette approche est également utilisée dans la passe de rendu.

Afin de maintenir une contribution lumineuse constante pour la lumière volumétrique continue utilisée dans LOA, il est important que chaque rayon ait une contribution unitaire. Depuis un schéma adaptatif d'échantillonnage utilisé, la contribution lumineuse  $l_i$  à chaque point d'échantillon  $i$ , doit être aussi adaptatif.

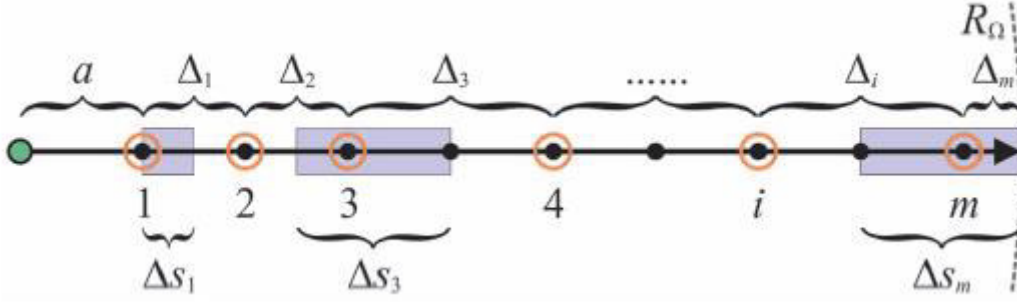


Figure 2.8. Illustration détaillée des distances d'échantillon le long d'un rayon LAO.

Ceci est résolu par déterminer l'intensité lumineuse en fonction de la taille des pas entre les emplacements d'échantillonnage actuels. La **figure 2.8** illustre une Rayon LAO avec échantillonnage adaptatif et quelques cas de contribution lumineuse ajoutée pour  $l_1$ ;  $l_3$  et  $l_m$ .

L'équation décrit le calcul de la contribution lumineuse pour chaque point d'échantillon:

$$l_i = \frac{\Delta s_i}{R_\Omega - \alpha'}, \quad \Delta s_i = \begin{cases} \Delta_1/2, & \text{if } i = 1 \\ \Delta_{i-1} + \Delta_i/2, & \text{if } 1 < i < m \\ \Delta_{i-1}/2 + \Delta_m, & \text{if } i = m \end{cases}$$

Avec une densité d'échantillonnage adaptatif, les opacités et les couleurs doivent également être corrigés, ce qui est traditionnellement fait en utilisant l'équation de correction d'opacité.

$$\alpha' = 1.0 - (1.0 - \alpha)^{Y^{\Delta_s/\Delta_B}}$$

Où  $\alpha'$  est l'opacité ajustée,  $\Delta_s$  représente l'étendue de l'échantillon, De plus, un paramètre  $Y$  est pour mieux contrôler l'apparence d'LAO.

La contribution de la lumière émissive dépend également de l'étendue des échantillons est ajusté en fonction de :

$$c'_E(s) = \frac{\Delta_s}{\Delta_B} c_E(s).$$

## 8. Application de la théorie de l'échantillonnage de Raycasting

Un ensemble de données CT peut être considéré comme un ensemble d'échantillons prélevés sur un signal continu à bande limitée. Selon le théorème d'échantillonnage de Nyquist-Shannon [17], la fréquence d'échantillonnage doit être au moins deux fois plus élevée que la fréquence la plus élevée du signal d'origine. Ainsi, pour une distance d'échantillon isotropique de  $T$  entre les voxels, la fréquence maximale  $\nu_f$  représentée dans l'ensemble de données est donnée par  $\nu_f = 1/2T$ . Le signal peut être parfaitement reconstruit en échantillonnant l'ensemble de données une fois par voxel, puis en effectuant une convolution avec un noyau de filtrage Sinc. Cependant, des erreurs supplémentaires sont introduites si le raycasting est utilisé pour reconstruire le signal.

La méthode de raycasting résout essentiellement l'intégrale de rendu en ajoutant les densités de volume le long de chaque rayon de visualisation. Généralement, une interpolation trilineaire est utilisée pour calculer les valeurs de densité aux points d'échantillonnage. D'un point de vue mathématique, cette approche correspond à une reconstruction avec un noyau de filtre linéaire «tente». Les erreurs qui se produisent en raison de la reconstruction avec le noyau de filtre sont appelées artefacts de filtrage. On note deux propriétés importantes de cette source d'erreur. Tout d'abord, les artefacts de filtrage ne peuvent pas être réduits en augmentant la fréquence d'échantillonnage. Si les échantillons sont toujours pris exactement au centre de chaque voxel, il n'y a aucun avantage à échantillonner le signal avec une fréquence plus élevée qu'une fois par voxel. Deuxièmement, contrairement au filtre Sinc, qui a un support infini, le support du filtre de tente est limité aux voxels adjacents. Ainsi, le taux d'échantillonnage requis dépend uniquement du comportement local de la densité du matériau.

C'est une propriété fondamentale qui permet l'utilisation de critères basés sur la fréquence locale pour contrôler l'échantillonnage adaptatif. Dans la discussion précédente, on a supposé que les points d'échantillon sont toujours placés au centre de chaque voxel. Cependant, pendant le raycasting, les échantillons sont pris le long du rayon de vision à différentes positions qui ne coïncident



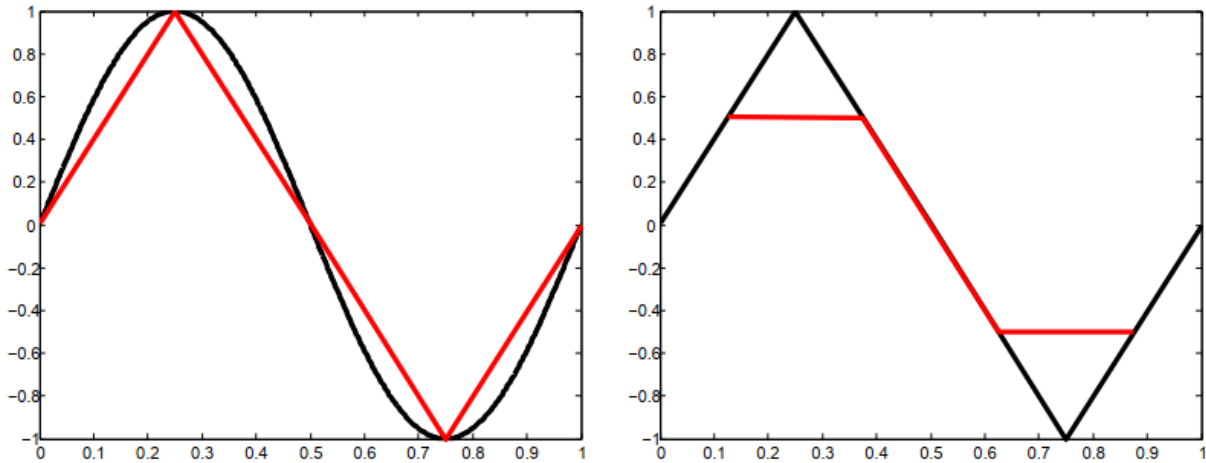


Figure 2.9. Un signal sinusoïdal (noir) est reconstruit avec une méthode de lancer de rayons (rouge).

La **figure 2.9** illustre l'erreur qui émane de cette procédure. La taille de l'erreur dépend de la fréquence du signal, de la fréquence d'échantillonnage et du point d'échantillonnage réel, qui ne peut être connu a priori. Afin de générer des images visuellement agréables et significatives, une fonction de transfert  $g$  est utilisée pour affecter des propriétés optiques aux valeurs de Hounsfield stockées dans l'ensemble de données CT [17]. Ainsi, la fréquence d'échantillonnage ne dépend pas seulement des valeurs de données  $f$ , mais plutôt de la fonction composite  $g \circ f$ . Ceci est particulièrement important si la fonction de transfert comporte des composants à haute fréquence. Dans ce cas, une fréquence d'échantillonnage élevée est nécessaire même si la variation de  $f$  est faible. Une méthode pour surmonter ce problème consiste à utiliser la pré-intégration comme précédemment introduit. Cependant, il faut noter que l'erreur d'échantillonnage discutée dans le paragraphe précédent peut être considérablement amplifiée par une fonction de transfert à haute fréquence.

Même avec l'utilisation de la pré-intégration, de petites erreurs dans l'échantillonnage de  $f$  peuvent conduire à de grandes erreurs pour la fonction composite  $g \circ f$ . Un bon critère d'échantillonnage basé sur la fréquence ne peut donc pas être basé sur l'analyse spectrale de  $f$  seul, mais plutôt sur l'analyse de  $g \circ f$ . Une analyse détaillée de l'influence de la fonction de transfert sur la fréquence d'échantillonnage. [17]

## 9. Analyse spectrale

La décomposition spectrale directe de la fonction composite  $(g \circ f)$  est une procédure intensive très computationnelle, Si la fonction de transfert ne change pas pendant le rendu, cette analyse peut être effectuée dans une étape de prétraitement. Cependant, dans les applications médicales, la fonction de transfert est habituellement modifiée pendant le rendu pour parcourir différentes structures anatomiques et régions d'intérêt. Dans cette section, on présente une approximation efficace pour la décomposition spectrale qui peut être évaluée avec très peu d'effort de calcul. Cela permet l'utilisation de critères d'adaptabilité basés sur la fréquence même pour la fonction de transfert qui change dynamiquement pendant le rendu.

### 9.2. Analyse spectrale de la fonction composite

Bergner et al. A montré que la fréquence maximale  $v_h$  de la fonction composite  $g(f(x))$  peut-être approchée par [17]:

$$v_h = v_g \max |f'(x)| \quad \text{Équation 2}$$

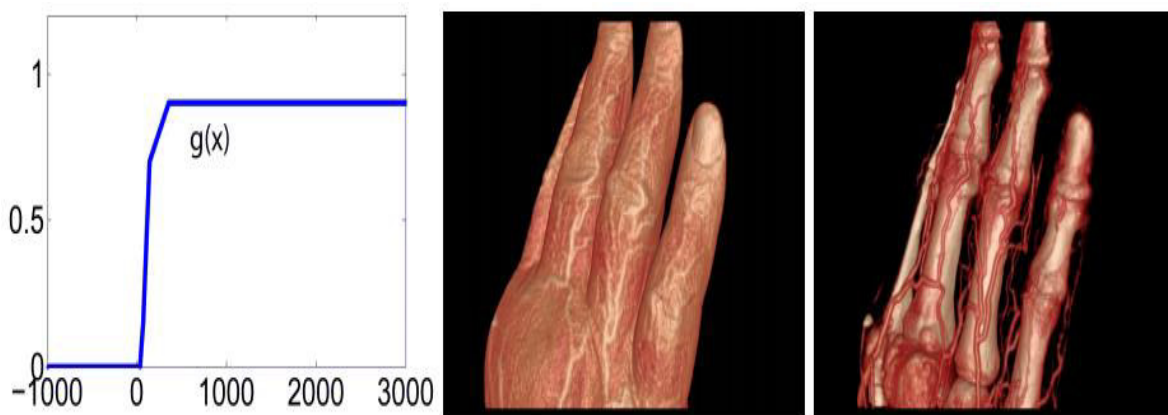
Où  $v_g$  désigne la fréquence maximale de  $g$ . Cette approximation n'est pas serrée, ce qui signifie que la fréquence maximale réelle pourrait être supérieure à l'estimation. Cependant, Bergner et al. A montré que l'énergie du signal décroît exponentiellement pour des fréquences supérieures à cette estimation. En accord avec ces résultats, nos expériences ont montré qu'aucun artefact visible n'apparaît même si aucun sur-échantillonnage additionnel n'est utilisé, Bien que l'estimation ci-dessus soit valable pour toutes les techniques de rendu de volume, elle peut être significativement améliorée pour la technique de raycasting en raison de la localisation du filtre de reconstruction. Lorsqu'on se rapproche de la fréquence pour déterminer la taille du pas du point d'échantillonnage  $s_1$  au point d'échantillonnage  $s_2$ , il est évident que le gradient maximal de  $f'(x)$  devrait être déterminé sur l'intervalle  $[s_1; s_2]$  et pas sur tout le volume.

De plus, on propose de déterminer la fréquence maximale de  $g$  sur toute la plage des valeurs de Hounsfield, mais plutôt sur ces valeurs  $I = [S_{\min}; S_{\max}]$  qui se produisent réellement dans  $[s_1; s_2]$ . Les avantages de cette approche deviennent rapidement évidents en examinant les fonctions de transfert généralement utilisées pour la visualisation médicale (voir la **figure 2.10**). Il existe de grandes zones où la fonction de transfert a des variations minimales alors qu'elle change sensiblement autour d'une certaine valeur de seuil. Donc peut être réduit si

toutes les valeurs de  $f$  dans  $[s1; s2]$  n'induisent que peu de changements dans la fonction de transfert. En utilisant l'approximation 1, la période d'échantillonnage est donnée par:

$$T = \frac{1}{2v_g(I) \cdot \max|f'(x)|} \quad \text{Équation 3}$$

Où  $v_g(I)$  est la fréquence maximale de  $g$  pour toutes les valeurs de  $I$ . Cette approximation peut être mise en œuvre très efficacement. Nous discuterons ensuite comment les fréquences maximales de  $g$  sont calculées.



*Figure 2.10. Une fonction de transfert typique d'opacité non stationnaire utilisé pour la visualisation médicale (gauche). La fonction de transfert est décalée le long de l'axe des  $x$  pour parcourir les différentes structures anatomiques d'une main humaine (milieu, droite)*

## 9.2. Fonctions de transfert dynamique

La décomposition spectrale de  $g$  est obtenue en appliquant une transformée de Fourier rapide (FFT). on définit la fréquence de coupure qui contient un pourcentage prédéfini de l'énergie du signal en tant que fréquence maximale de  $g$ . Si la fonction de transfert change arbitrairement pendant le rendu, la FFT doit être appliquée à nouveau. Bien que la FFT soit un algorithme très efficace, cela ne peut pas être fait en temps réel pour des fonctions de transfert à haute résolution. Pour surmonter ce problème, nous utilisons le fait que dans les applications médicales, la fonction de transfert n'est modifiée que de manière très restreinte lors de l'interaction de l'utilisateur.

Les postes de travail d'imagerie médicale, permettent généralement à l'utilisateur de changer une fonction de transfert 1D avec de simples gestes de souris. Afin de mettre en évidence différentes structures anatomiques (voir **figure 2.10**), les quatre composantes RGBA de la

fonction de transfert  $g(x)$  sont décalées le long de l'axe des  $x$  ou multipliées par un facteur constant. Ce comportement peut également être modélisé en définissant la carte affine.

$$t \rightarrow \psi(t) = \alpha\psi t + b\psi \quad \text{Équation 4}$$

Et en utilisant cette carte pour ajuster la densité de matériau  $f$  avant que la fonction de transfert  $g$  ne soit appliquée. Pour déterminer la fréquence maximale d'une fonction de transfert modifiée, nous devons analyser la composition de la fonction  $g \circ \psi \circ f$  au lieu de  $g \circ f$ . En notant que:

$$(\psi \circ f)' = (\alpha_\psi f + b_\psi)' = \alpha_\psi f' \quad \text{Équation 5}$$

On peut définir la nouvelle estimation:

$$T = \frac{1}{2v_g(I) \cdot |\alpha_\psi| \cdot \max |f'(x)|} \quad \text{Équation 6}$$

Incorporer le changement de la fonction de transfert dans l'approximation. Cela permet d'approcher la fréquence maximale des fonctions de transfert modifiées avec un effort de calcul minimal en utilisant la décomposition spectrale de la fonction de transfert d'origine. Une FFT sur  $g$  ne doit être effectuée que si une nouvelle fonction de transfert est choisie, mais pas si la fonction de transfert est modifiée de la manière décrite ci-dessus. Il est important de remarquer que cela correspond parfaitement au cas d'utilisation typique de la visualisation médicale. Une fonction de transfert entièrement nouvelle est habituellement choisie uniquement au début de l'examen après le chargement de l'ensemble de données. A ce stade, le retard causé par le calcul de la FFT est négligeable. Le changement de la fonction de transfert pendant le rendu interactif peut alors être modélisé comme décrit ci-dessus.

### 9.3. Implémentation

L'algorithme de lancer de rayon a été implémenté en tant que module pour la boîte à outils de visualisation (VTK) en utilisant les langages de programmation C++ et OpenGL / GLSL. On utilise la pré-intégration et stockons les données pré-calculées nécessaires dans une texture 2D. Afin de mettre en œuvre l'échantillonnage adaptatif en utilisant le critère basé sur la fréquence, deux textures supplémentaires sont utilisées. Une texture 2D contient la fréquence maximale  $v_g(I)$  de  $g$  dans l'intervalle  $I = [s_{\min}; s_{\max}]$ . Comme détaillé ci-dessus, ces fréquences peuvent être calculées dans une étape de pré-traitement en effectuant une décomposition spectrale de  $g$  en utilisant la FFT.

Les gradients  $f'(x)$  sont stockés dans un volume d'importance qui est de la même taille que l'ensemble de données. Pour chaque voxel, le gradient de  $f$  est déterminé dans chaque direction spatiale. Il est important que le critère d'adaptabilité évalue la fréquence maximale de la région entière couverte par la prochaine étape d'échantillonnage. Pour ce faire, les gradients sont filtrés avec un filtre max avec une largeur de  $2T_{max}$  et le résultat est stocké dans le volume d'importance. Ici,  $T_{max}$  indique la taille de pas maximale. Veuillez noter que cette procédure implémente essentiellement un critère d'échantillonnage isotrope. Une accélération supplémentaire pourrait être obtenue si la direction du rayon était prise en compte.

Afin d'estimer la taille du pas, nous avons également besoin des valeurs  $s_{min}$  et  $s_{max}$  dans la région couverte par le rayon lors de la prochaine étape d'échantillonnage. Bien qu'il soit possible de stocker ces valeurs dans un volume d'importance similaire aux gradients  $f'(x)$ , cette approche nécessiterait beaucoup de mémoire. Pour contourner ce problème, on approche  $f$  avec son expansion de Taylor.

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \quad \text{Équation 7}$$

Autour du point d'échantillonnage  $x_0$ . Les bornes supérieure et inférieure pour  $s_{min}$  et  $s_{max}$  peuvent ensuite être calculées en utilisant les relations

$$s_{max} = f(x_0) - f'(x_0)T_{min} \quad \text{Équation 8}$$

$$s_{max} = f(x_0) + f'(x_0)T_{max} \quad \text{Équation 9}$$

Et les gradients stockés dans le volume d'importance.

On conclure la description de l'implémentation en résumant l'effort de calcul supplémentaire associé à la méthode adaptative. L'algorithme nécessite des opérations en virgule flottante supplémentaires pour calculer l'intervalle  $I = [s_{min}; s_{max}]$  valeurs selon l'équation 9. Du fait que la fonction de transfert change pendant le rendu, l'intervalle doit être décalé comme décrit par l'équation 3. Enfin, l'évaluation de la taille du pas en utilisant l'approximation fréquentielle (voir Équation 5) doit être effectuée. Deux extractions de texture supplémentaires doivent être effectuées pour obtenir le gradient  $f'$  et la fréquence maximale locale de  $v_g$ . Plus important encore, la méthode adaptative utilise plus de mémoire de texture pour stocker le volume d'importance. Cette implémentation utilise un volume d'importance de la même taille que l'ensemble de données d'origine. Cependant, il convient de souligner qu'il

contient beaucoup de redondance en raison de l'utilisation du filtre max et peut donc être facilement compressé.

## **10. Conclusion**

Dans ce chapitre on a présenté le raycasting adaptatif Ainsi que quelques-unes des méthodes qui utilisent la même technique. Le chapitre suivant discutera la conception et l'implémentation de notre projet.

## **1. Introduction**

Ce chapitre se compose par de deux parties : la conception et l'implémentation de notre projet. La conception consiste à déterminer et schématiser les grandes lignes des mécanismes qui devront être programmés en vue d'obtenir chacune des fonctions que devra offrir le logiciel. Des plans conceptuels du logiciel selon les formalismes de modélisation seront alors réalisés. De l'autre côté l'implémentation consiste à la rédaction du code source, des instructions de programme qui offriront les fonctions attendues, et qui sont le corps du logiciel.

Dans l'implémentation, l'environnement de développement matériel et logiciel utilisé pour réaliser notre application est révélée, puis on va exposer quelques tests et effectuer une analyse de résultats.

## **2. Motivation et Objectif**

L'Algorithme de Ray casting est une technique de rendu utilisée en infographie et en géométrie computationnelle. Il est capable de créer une perspective tridimensionnelle dans une carte en deux dimensions. Il est utilisé des tests d'intersection rayon-surface pour résoudre une variété de problèmes dans l'infographie et la géométrie informatique.

Il a beaucoup de domaines d'applications: Prévision météo, physique et l'imagerie médicale. Notre travail présente l'imagerie médicale. Les médecins ont besoin d'avoir une meilleure vision des données pour un diagnostic plus rapide et plus précise surtout pour certain zone pour la détection des maladies pour avoir le traitement approprié. Pour cela nous suggérons un algorithme de raycasting adaptatif pour remédier et avoir une visualisation précise par rapport l'algorithme de raycasting de base dans les zones d'intérêt.



### 3. Conception

Cette section est la conception de notre système.

#### 3.1. Conception globale

Nous présentons cette architecture pour la réalisation de notre application, cette dernière consiste à la lecture d'ensemble de tranches 2D (slices). A partir de cet ensemble, on fait la construction d'une grille régulière. Puis nous allons définir le couleur et opacité qui convient chaque densité avec (définition les propriétés de fonction de transfert), ensuite un type de raycasting (MIP ou par accumulation) va être effectué pour la visualisation du volume.

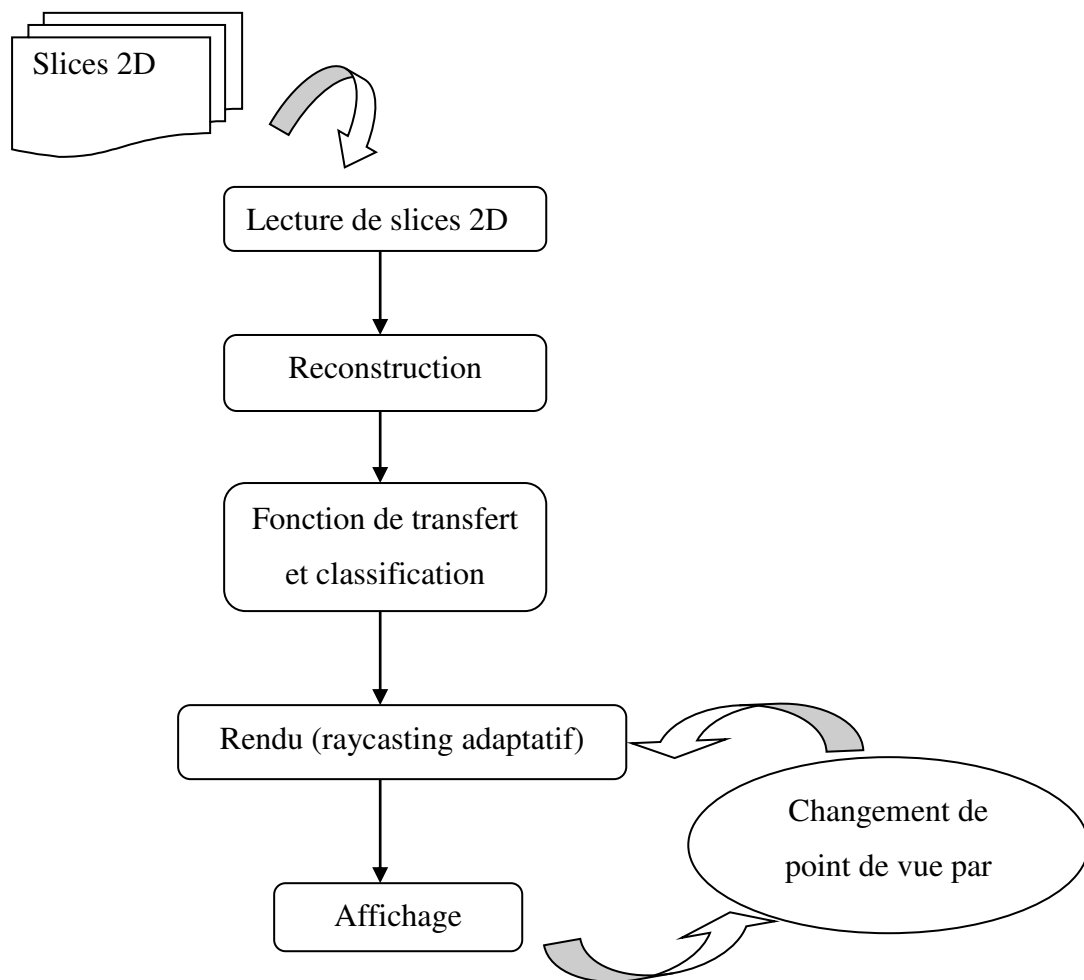


Figure 3.1 Architecture global

### **3.1.1. Reconstruction**

C'est le module responsable de la reconstruction d'un volume à partir de l'ensemble des tranches 2D (slices) après la lecture de ce dernier, cette reconstruction va produire un volume composé d'un ensemble de voxels ou chaque voxel est défini par son position et son densité.

### **3.1.2. Fonction de transfert et classification**

C'est le module de configuration ou bien l'association d'une couleur et d'une opacité à chaque densité, ce qu'est un vrai défis, il est pratiquement très difficile d'assigner les bonnes valeurs d'opacité et de couleur pour représenter une densité donnant la variété des valeurs de densités équivalentes aux parties du corps humain (os, tissus mous, sang, poumons ... etc.), des valeurs erronées peuvent rendre des parties du corps humain qui doit être opaques (comme les os) transparentes ou l'inverse, la meilleure approche pour résoudre ce problème est d'essai et d'erreur jusqu'à la réalisation des résultats satisfaisants.

Pour améliorer la qualité du rendu nous utilisons l'opacité de gradient. Elle est calculée en multipliant l'opacité correspondante à la valeur scalaire par un multiplicateur d'opacité qui est représentée par amplitude de gradient de données scalaires. Ce multiplicateur est utilisé pour renforcer la contribution des régions de transition du volume.

### **3.1.3. Rendu (Raycasting adaptatif)**

C'est le module qui rendre la visualisation possible, d'abord, il applique la technique de raycasting. Le raycasting est un algorithme de rendu volumique direct qui appartient à la catégorie des méthodes backwards (ordre image), il s'agit de lancer un rayon à chaque pixel de l'écran vers le volume, il faut citer que la méthode de projection utilisée dans notre application est la projection parallèle et le point de vue initial est le centre de l'écran.

### **3.1.4. Affichage (visualisation de volume)**

Le module final de l'architecture est le processus d'interprétation des valeurs des couleurs et opacités retournées par le rendu à un volume semi-transparent coloré affiché par l'écran. L'utilisateur peut appliquer des interactions avec l'application (zoom, rotation), ces interactions vont changer le point de vue et ainsi recalculer le rendu (le module précédent) avec le nouveau point de vue.

## 4. Conception détaillée

### 4.1. Introduction

Dans cette partie on va détaillée la conception de notre système, cette partie composée de deux phase, la première présente le raycasting statique et la deuxième présente le raycasting adaptatif.

### 4.2. Raycasting statique :

L'architecture détaillée de ray casting statique peut être schématisé par la manière suivante :

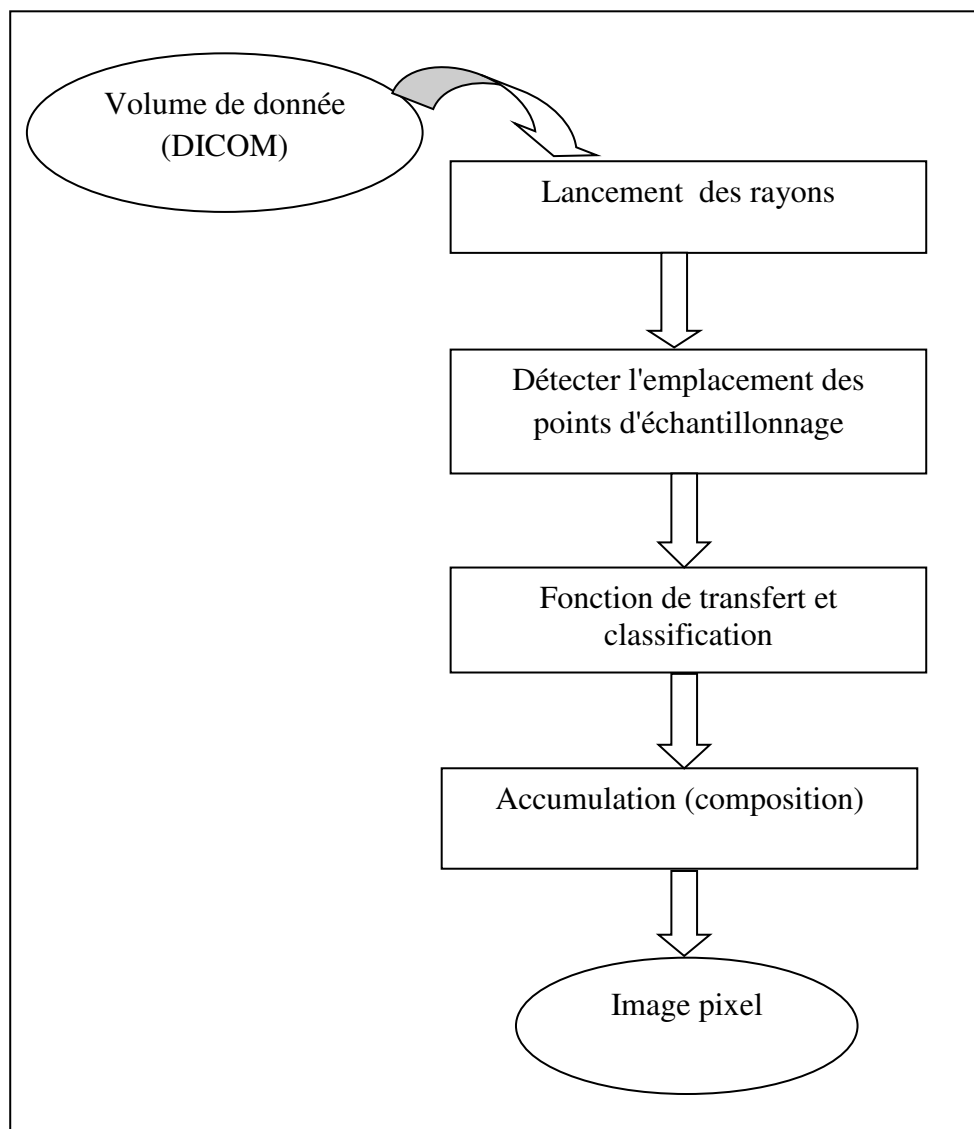


Figure 3.2. L'architecture détaillée du Ray Casting statique

➤ **Volume de donnée :**

Les données volumiques sont des entités 3D (éventuellement variables dans le temps) pouvant contenir des informations à leur intérieur. Les relations de voisinage définies entre les points constituent la topologie de la grille structurée ou non structurée. La définition d'une grille régulière est liée à la notion de voxel.

➤ **Lancement des rayons :**

Un rayon virtuel est lancé explicitement dans le volume pour chaque pixel de l'espace de l'image et l'intégration est calculée le long de ce rayon par échantillonnage à intervalles réguliers.

➤ **Détecter l'emplacement des points d'échantillonnage :**

L'échantillonnage, c'est-à-dire la reconstruction de l'ensemble de données volumiques est couramment effectuée en utilisant une interpolation.

➤ **Accumulation (composition) :**

Les couleurs interpolée et l'opacité sont ensuite fusionnées avec les autres et avec l'arrière-plan en les combinant dans un ordre avant-arrière ou arrière-plan pour obtenir la couleur du pixel. (L'ordre de composition ne fait aucune différence dans l'image de sortie finale.).

L'algorithme détaillé de l'algorithme du raycasting statique peut être résumé comme suit :

- 1: Déterminer la position d'entrée de volume
- 2: Calculer la ligne et la direction du rayon
- 4: tant que (Position actuelle du rayon dans le volume) faire
- 5: interpoler la valeur scalaire à la position actuelle
- 6: Classer la valeur scalaire avec la fonction de transfert
- 7: Composition de la couleur et de l'opacité
- 8: Calculer la position de l'échantillon suivant le long du rayon
- 9: fin tant que

Pour chaque pixel, la première étape consiste à calculer le point d'intersection entre le rayon et le volume. Si le rayon n'intersecte pas le volume, le pixel est défini sur la couleur de fond. S'il y a une intersection, alors l'étape suivante consiste à extraire l'intensité interpolée du voxel au point d'intersection. L'intensité est ensuite convertie en une valeur RVB en utilisant une fonction de transfert de couleur. La valeur RVB est combinée avec une opacité de 1,0 pour produire la valeur RVBA finale du pixel. On effectue par la suite le ré-échantillonnage ou composition de la couleur et de l'opacité.

### 4.3. Raycasting adaptatif

L'architecture détaillée de raycasting adaptatif peut être schématisé par la manière suivante :

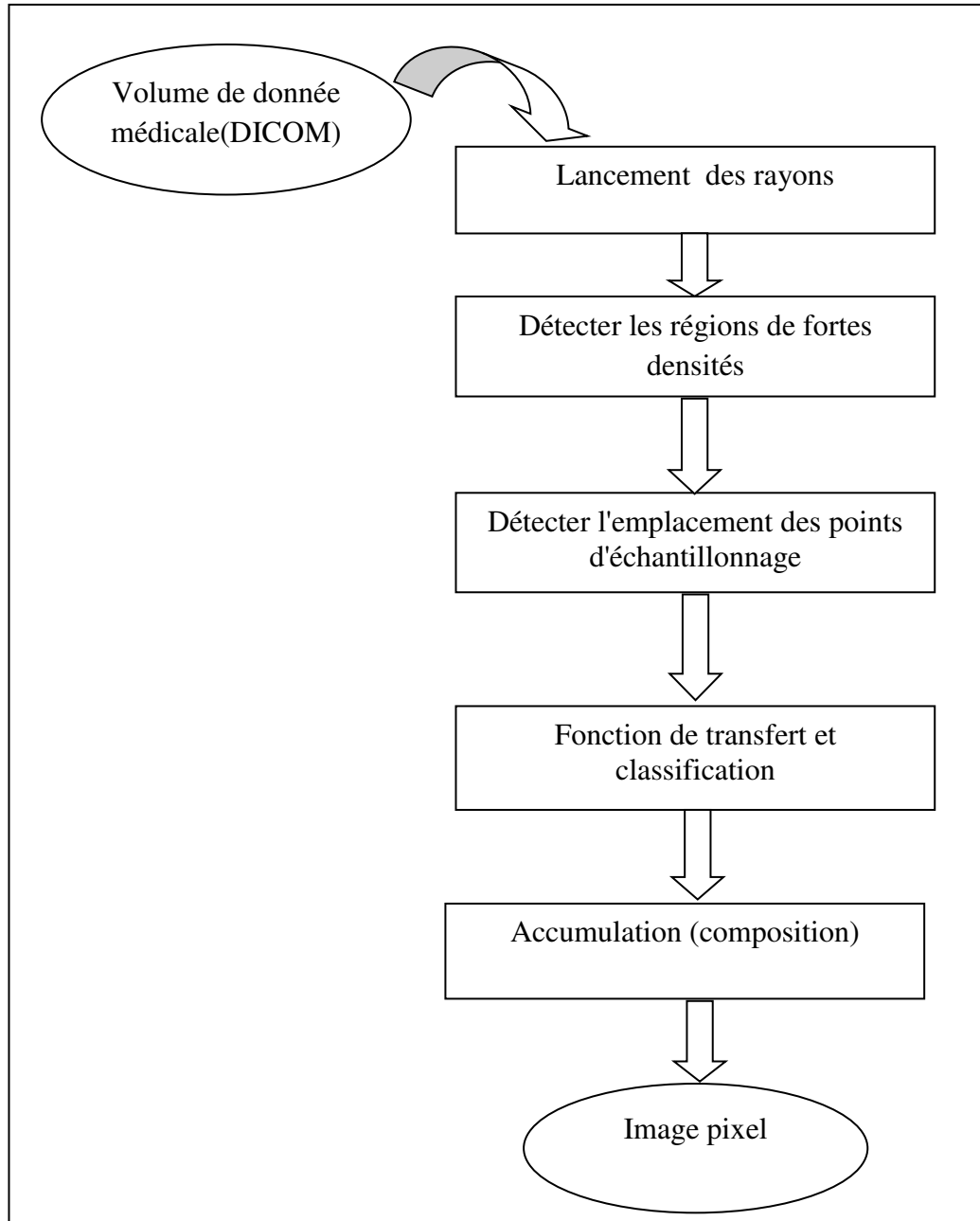
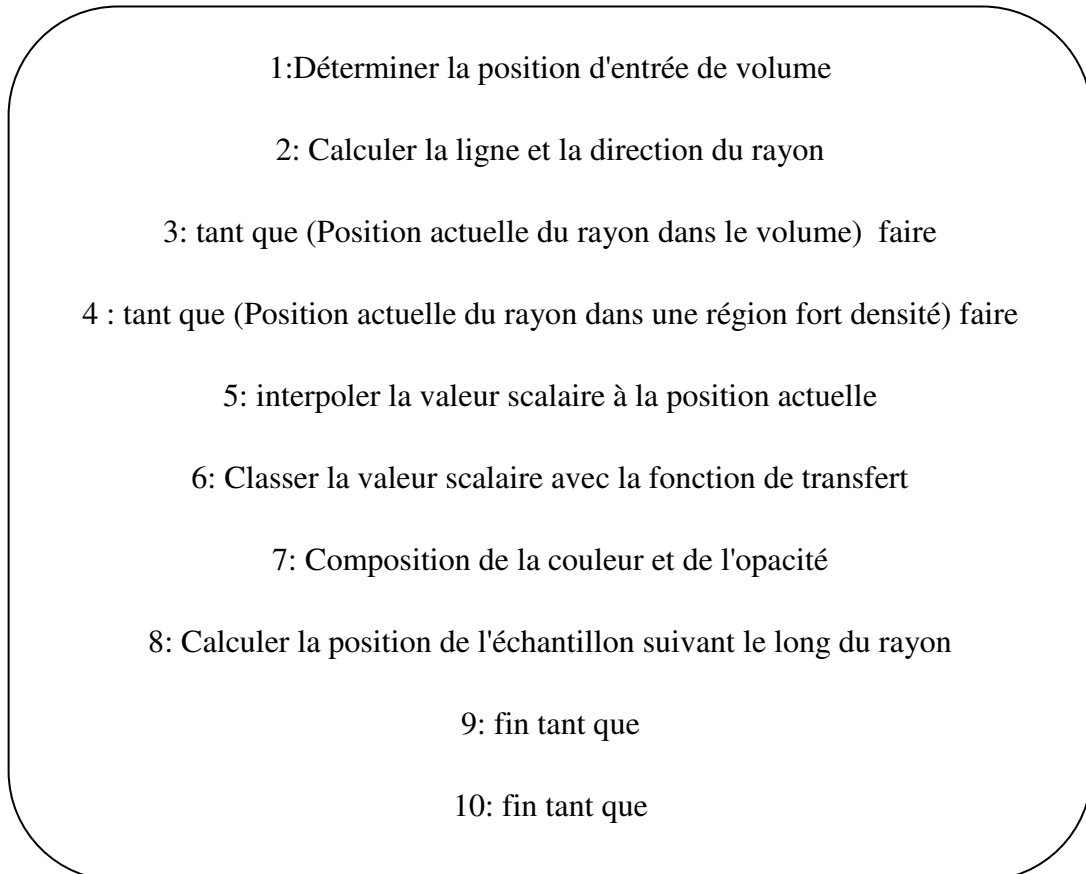


Figure 3.3. L'architecture détaillée du Ray Casting adaptatif

➤ **Détecter les régions de fortes densités**

L'échantillonnage sont prélevés par un grand nombre des échantillons dans des zones où la densité est très élevée, c.-à-d. ignorés les régions vides.

L'algorithme détaillé de l'algorithme du raycasting adaptatif peut être résumé comme suit :



Pour détaillé l'étape de choix des régions dense c.-à-d. les régions ou la variation des couleurs est élevé, on va suivre la fonction « ComputeRayInfo » en raycasting statique et comment on peut faire une modification pour rendre la fonction adaptatif avec les volumes données.

La fonction « ComputeRayInfo » Dans la méthode statique peut être résumée comme suit :



1 : Début : Entrée :

2 : ER=point d'entrée du rayon dans le volume.

3 : SR =point de sortie du rayon dans le volume.

4 : Dir = direction du rayon ; Pos = position du rayon ; BP = Boucle de pas.

5 : VF = valeur final du rayon ; NP = Nombre de pas.

6:Calculer la valeur X du point de vue du rayon.

7: Calculer la valeur Y du point de vue du rayon.

8: Transforme ce (X ; Y) point avec une valeur Z de 0 à 1.

9 : Si (pas > 0) Alors

10: ER= Original ER + (pas\*Dir).

11: Finsi.

12: Si (ER > 0) Alors

13: définir la position.

14: définir la direction.

15: Finsi

16: Pour (BP = 0 ; BP < 3 ; ++) Faire

17 : Si (VF > Pos [BP]) Alors

18 : Actuel pas = 1+ (VF – pos [BP]/Dir [BP]).

19 : Sinon Actuel Pas = 0.

20 : Finsi.

21 : Si (Pos [BP] > VF) Alors

22 : Actuel pas = 1+ (Pos [BP] - VF /Dir [BP]).

23 : Sinon Actuel Pas = 0.

24 : Finsi

25 : Si (Actuel Pas < NP) Alors

26 : NP = Actuel pas.

27 : Finsi

28 : Fin

La fonction « ComputeRayInfo » Dans la méthode Adaptatif peut être résumée comme suit :

1 : Entrée :

2 : ER=point d'entrée du rayon dans le volume.

3 : SR =point de sortie du rayon dans le volume.

4 : Dir = direction du rayon ; Pos = position du rayon ; BP = Boucle de pas.

5 : VF = valeur final du rayon ; NP = Nombre de pas.

6 : Début : Calculer la valeur X du point de vue du rayon.

7: Calculer la valeur Y du point de vue du rayon.

8: Transforme ce (X ; Y) point avec une valeur Z de 0 à 1.

9 : Si (pas > 0) Alors

10: ER= Original ER + (pas\*Dir).

11: Finsi.

12: Si (ER > 0) Alors

13: définir la position.

14: définir la direction.

15: Finsi

16: Pour (BP = 0 ; BP < 3 ; ++) Faire

17 : Si Pos [BP] !vide Alors

18 : Si (VF > Pos [BP]) ET Pos [BP] != Pos [précédente]) Alors

19 : Actuel pas = 1+ (VF – pos [BP]/Dir [BP]).

20 : Sinon Actuel Pas = 0.

21 : Finsi.

22 : Si (Pos [BP] > VF) ET (Pos [précédente]) Alors

23 : Actuel pas = 1+ (Pos [BP] - VF /Dir [BP]).

24 : Sinon Actuel Pas = 0.

25 : Finsi

26 : Si (Actuel Pas < NP) Alors

27 : NP = Actuel pas.

28 : Finsi ; 29 : Fin

Pour éviter que le rayon prenne deux ou plus des échantions similaires, ou bien homogène, avec l'échantion précédent, on a proposé la fonction suivant :

1 : Fonction Précédent  
2 : Entrée :  
3 : Dir = direction du rayon ; Pos = position du rayon ; BP = Boucle de pas.  
4 : Col= colleur d'échantion ; NPI = Nombre de pas initial.  
5 : NPF = Nombre de pas final.  
6 : Début  
7 : Calculer la valeur X du point de vue du rayon.  
8: Calculer la valeur Y du point de vue du rayon.  
9: Transforme ce (X ; Y) point avec une valeur Z de 0 à 1.  
10 : Tant que NPF >NPI faire  
11: Si (Dir [0] = Dir [1]) et (Pos [0] != Pos [1]) Alors  
12 : Si Col [1] = Col [0] Alors  
13 : BP++ ;  
14 : Sinon Calculer Col [1] ;  
15: Finsi.  
16: Sinon // (Dir [0] != Dir [1]) et (Pos [0] != Pos [1]) Alors  
17 : Si Col [1] = Col [0] Alors  
18 : BP++ ;  
19 : Sinon Calculer Col [1] ;  
20 : Finsi.  
21 : Finsi.  
22 : Fin

## **5. Implémentation**

La deuxième partie de ce chapitre est l'implémentation, c'est la partie d'application de notre vision conceptuelle afin de réaliser notre application.

### **5.1 Environnement de l'application**

La mise en œuvre de n'importe quelle application requiert un environnement de développement matériel et logiciel et la nôtre est définie de la manière suivante :

#### **5.1.1. Environnement de développement matériel**

Pour l'implémentation de notre application nous avons utilisés une machine PC avec les caractéristiques suivantes :

- ❖ Processeur : Intel ® Core (TM) i5 CPU M 560 @ 2.67 GHZ 2.67 GHZ
- ❖ RAM : 4 GB
- ❖ Carte graphique: Intel(R) HD Graphics
- ❖ Disque dur : 235 GO

#### **5.1.2. Environnement de développement logiciel**

Notre application a été implémentée sous un système d'exploitation Windows 7 32-bits, plusieurs outils de programmation ont été utilisé, nous avons choisi Microsoft Visual Studio 2010 comme un IDE (Environnement de développement intégré) et le C++ comme un langage de programmation pour bénéficier de sa paradigme orientée-objet. L'intégration d'une plateforme de programmation haute niveau graphique (Visualisation ToolKit) a été nécessaire pour rendre note implémentation possible.

La visualisation Toolkit (VTK) est un système logiciel open-source gratuitement disponible pour l'infographie 3D, le traitement et la visualisation d'image. VTK se compose des bibliothèques des classes C++ et de plusieurs couches d'interface interprétées notamment de Tcl / Tk, Java et Python. VTK prend en charge une grande variété d'algorithmes de visualisation des données scalaires, vectoriels, tenseurs et textures, Il offre des techniques de modélisation avancées telles que: la modélisation implicite, réduction de polygones, maillage, lissage, coupe, courbes de niveau, et la triangulation de Delaunay. VTK est multiplateforme et fonctionne sur Linux, Windows, Mac et Unix.

## **5.2. Le type d'images médicales traité**

Les images (coupes / tranches/ slices) utilisées dans notre application sont de format DICOM.

DICOM - Digital Imaging and Communications in Medicine - est la norme internationale pour les images médicales et d'informations connexes (ISO 12052). Il définit les formats d'images médicales qui peuvent être échangés avec les données et la qualité nécessaire pour une utilisation clinique.

DICOM est mis en œuvre dans presque tous radiologie, imagerie de cardiologie et dispositifs de radiothérapie (rayons X, CT, IRM, échographie... etc.), et de plus en plus dans des dispositifs dans d'autres domaines médicaux tels que l'ophtalmologie et la dentisterie. Avec des dizaines de milliers de dispositifs d'imagerie en cours d'utilisation, DICOM est l'une des normes de messagerie les soins de santé les plus largement déployées dans le monde. Il ya littéralement des milliards d'images DICOM actuellement utilisés pour les soins cliniques.

Depuis sa première publication en 1993, DICOM a révolutionné la pratique de la radiologie, permettant le remplacement d'un film X -ray avec un flux de travail entièrement numérique. Tout comme l'Internet est devenu la plate-forme pour de nouvelles applications de l'information des consommateurs, DICOM a permis des applications avancées d'imagerie médicale qui ont "changé le visage de la médecine clinique". De l'urgence à des tests de stress cardiaque ou la détection du cancer du sein, DICOM est la norme qui rend le travail d'imagerie médicale. [18]

## **5.3 Tests et Analyse de résultats**

Des différentes CT des différentes parties de corps humain sont utilisés dans ces tests, on a obtenu des résultats variés en termes de qualité et temps de rendu. Les tests sont faits sur un ensemble des images (slice) de format DICOM, les figures suivantes représentent un CT des pieds d'un patient anonyme, la résolution de chaque tranche (slice) est 512 x 512, le nombre de tranches est 250 avec des méthodes visualisation et types d'interpolation variées. La fonction de transfert utilisée est la même dans tous les tests, c'est notre effort propre pour trouver les valeurs les plus adéquates aux parties de corps humain. [18]

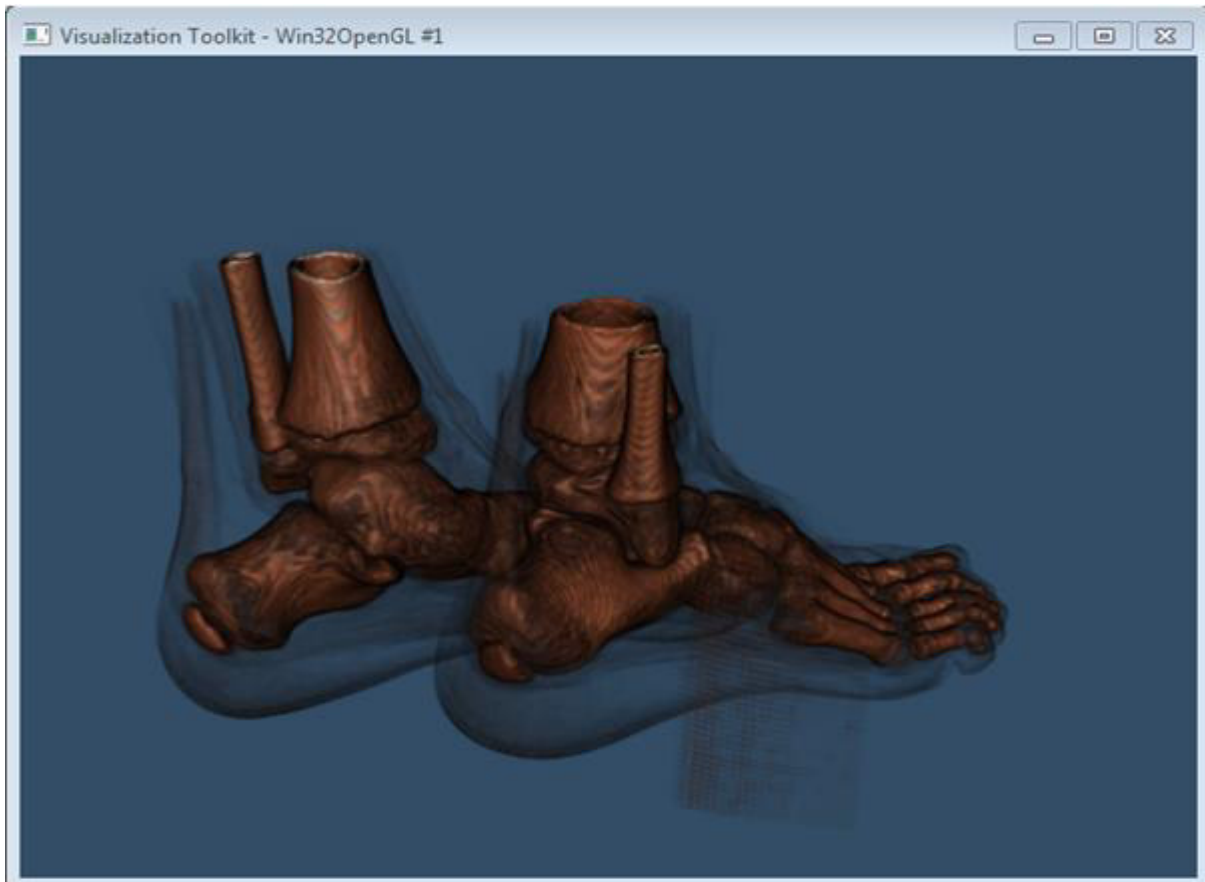


Figure 3.4. Raycasting des pieds, temps de rendu 1.32, distance d'échantionage: 5

La figure suivante représente le rendu volumique par le raycasting du même ensemble d'image avec Zoom

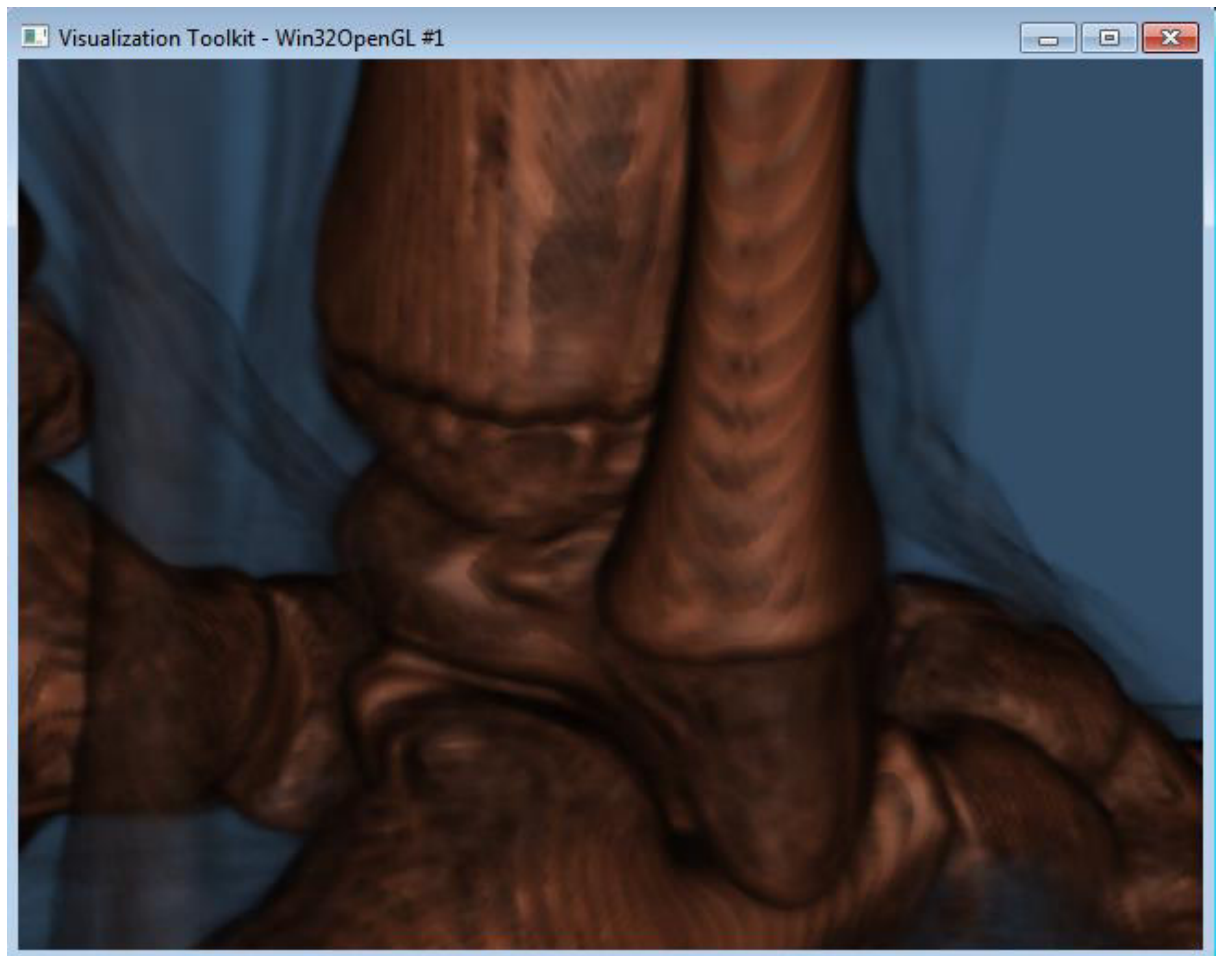


Figure 3.5. Raycasting des pieds, temps de rendu 7.14, distance d'échantonnage: 5

La figure suivante représente le rendu volumique par le raycasting du même ensemble d'image avec une rotation.



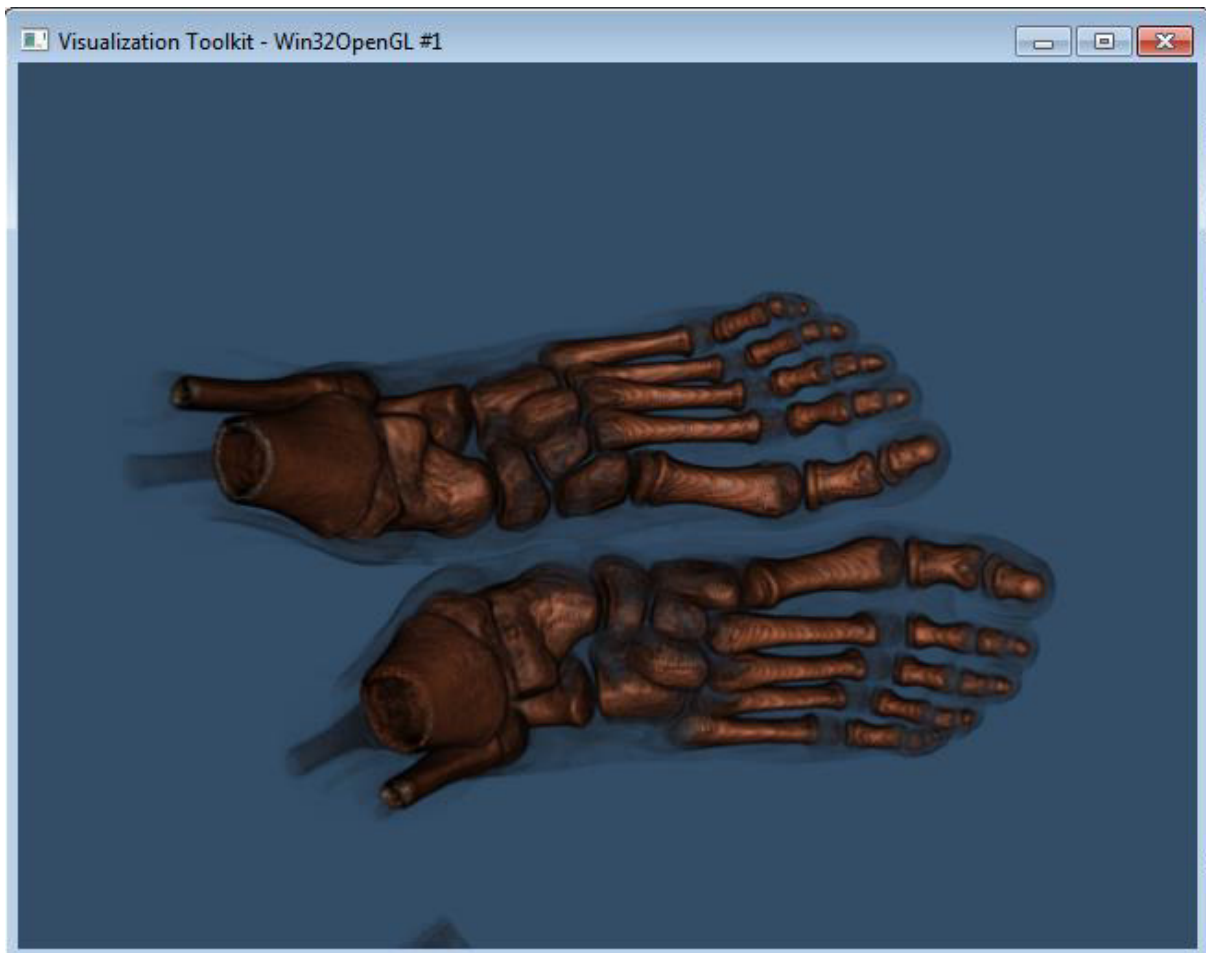
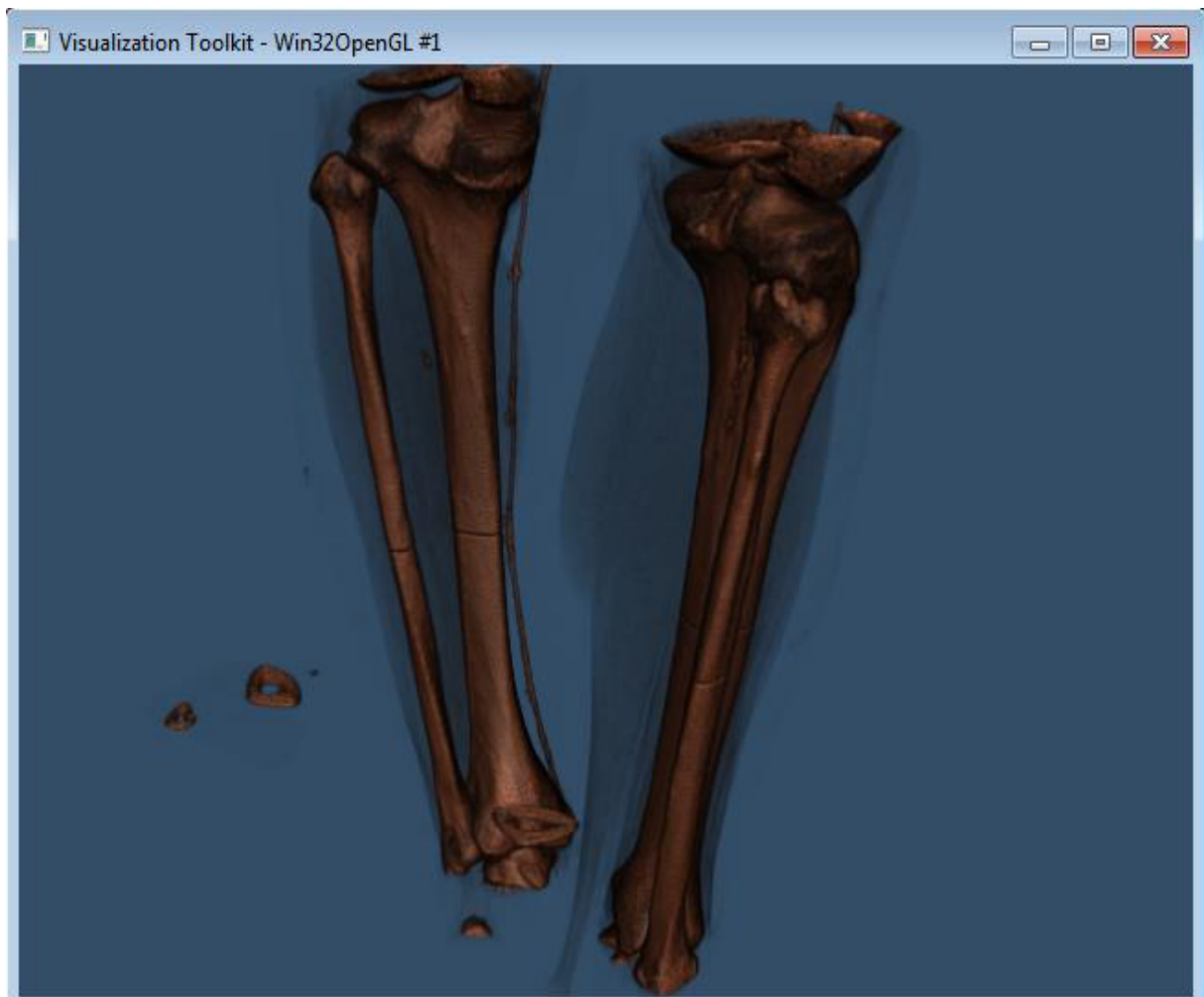


Figure 3.6. Raycasting des pieds ,temps de rendu 6.41 ,distance d'échantionnage: 5

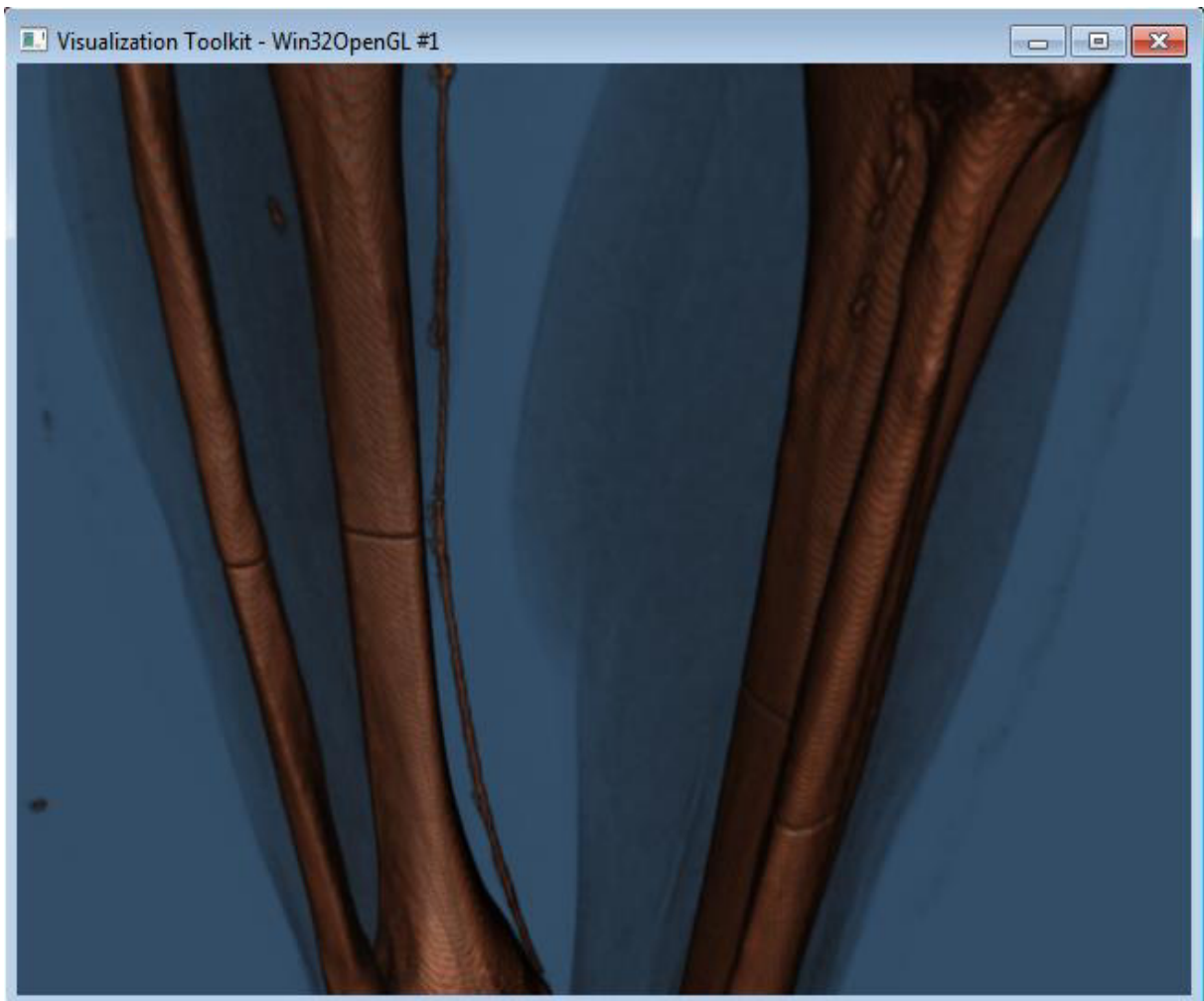
Maintenant on va présenter l'affichage d'un autre ensemble d'image avec les trois positions (rotation, profondeur, position initial).

La figure suivante représente le rendu volumique par le raycasting de la position initial.



**Figure 3.7. Raycasting, temps de rendu 0.33, distance d'échantionnage: 5**

La figure suivante présenter le rendu volumique de la même ensemble des images avec un Zoom



**Figure 3.8. Raycasting, temps de rendu 4.01, distance d'échantionnage: 5**

La figure suivante présenter le rendu volumique du même ensemble des images avec une rotation.

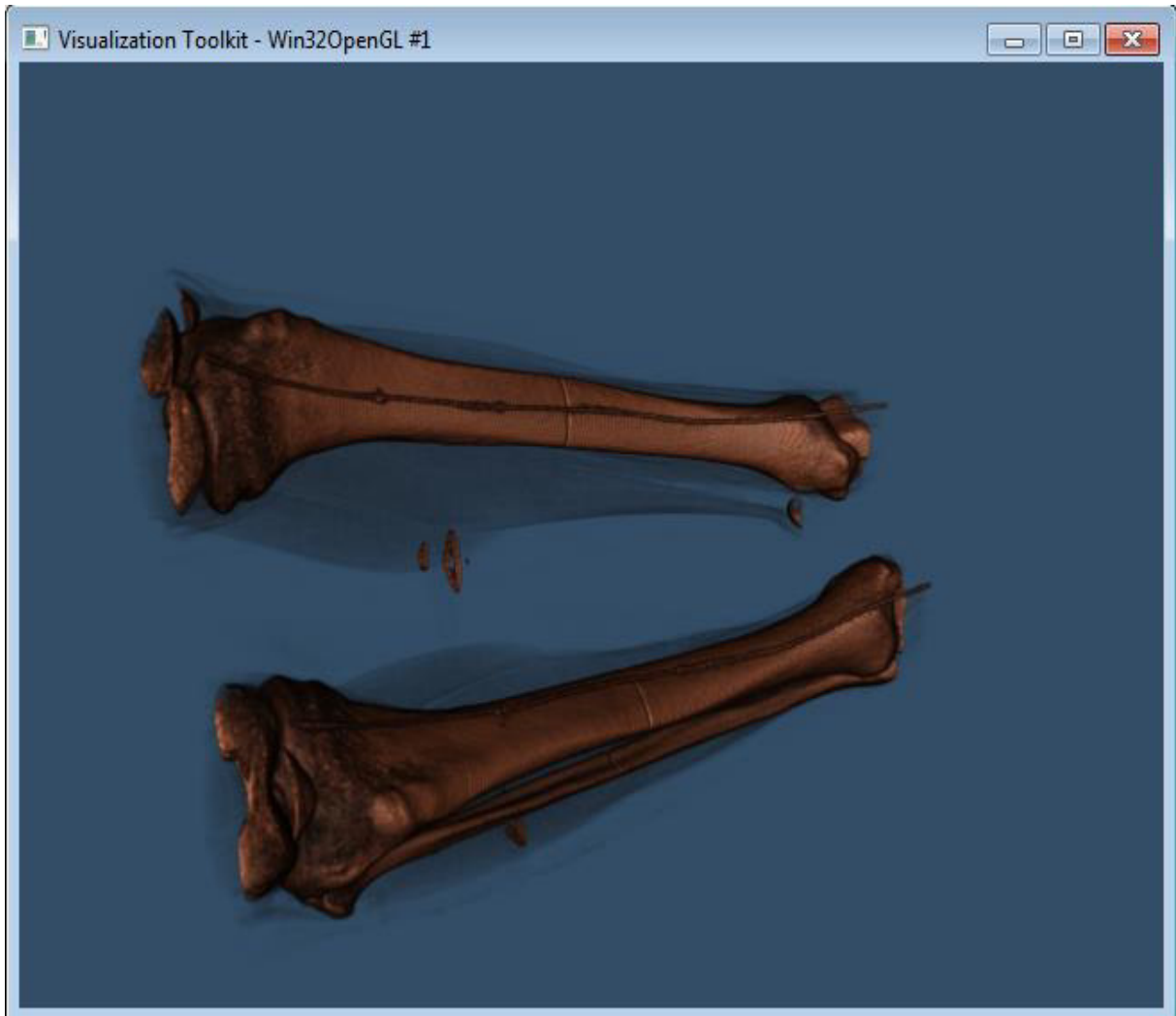


Figure 3.9. Raycasting des pieds, temps de rendu 10.61, distance d'échantonnage: 5

Pour démontrer l'importance de la visualisation volumique par raycasting, il faut repérer la différence entre la visualisation des tranches 2D et la visualisation de rendu volumique direct en fonction de faciliter de compréhension et de quantité d'informations affichées. Il suffit de comparer les deux images dans la figure suivante (3.10) qui représente le même CT visualisé par le raycasting par accumulation et par la visualisation traditionnelle des tranches 2D.



**Figure 3.10.** Comparaison entre Visualisation volumique par le Raycasting (gauche) et visualisation traditionnelle des tranches 2D

## **6. Conclusion**

Nous avons présenté dans ce chapitre la mise en œuvre de notre application qui permet de faire une visualisation volumique où on a utilisé la technique de ray casting qui travaillé sur le contenu de volume de telle façon elle lance des rayons pour obtenir des échantillons qui permet d’avoir une visualisation volumique.

## Conclusion générale

Dans ce projet on a vu le rendu volumique par raycasting et raycasting adaptatif. La structure de notre travail est composée de trois chapitres : l'état de l'art, raycasting adaptatif et la conception et l'implémentation.

Le premier chapitre couvre les différentes connaissances sur la visualisation scientifique qui fait une grande partie de l'infographie. Nous avons exploré profondément le rendu volumique par raycasting, ensuite, nous avons présenté dans le second chapitre les méthodes de raycasting adaptatif et les recherches dans le contexte après la troisième chapitre on présente la conception de notre application et l'implémentation qui est basée sur la précision de l'environnement de développement matériel, le standard DICOM, et logiciel utilisé et pour mettre en œuvre notre application et des résultats obtenus.

A travers ce mémoire, il apparaît très clairement que le rendu volumique direct est un processus complexe. Plusieurs étapes sont nécessaires avant d'obtenir une visualisation 3D.

L'objectif dans notre méthode de raycasting adaptatif est d'améliorer le rendu final d'une série d'image en 3D, par réduire et éliminer les points d'échantillonnages dans les régions vides et les régions homogènes, et augmenter l'échantillonnages dans les régions qui ont une grande variation des couleurs très élevé (les régions dense).de cette manière on développé la méthode de raycasting statique qui a fait un échantillonnages statique sans prend en compte le contenu de volume et comment choisi l'échantillonnages.

Nous avons testé notre application sur un ensemble des séries DICOM d'images médicales de différents organes humains.

## Introduction générale :

L'informatique est un outil de plus en plus utilisé par les scientifiques, car elle offre un moyen rapide et efficace de manipuler et d'extraire de l'information de données scientifiques. Parmi les moyens d'extraction d'informations de données, on trouve notamment la visualisation, qui consiste à offrir à un utilisateur une représentation d'un ensemble de données numériques. Cette représentation peut prendre différentes formes, dont un texte ou une image. Son principal but est de proposer un moyen d'interpréter correctement des données en offrant à la fois la représentation la plus intuitive possible et en se fiant et s'adaptant à la perception humaine, la vue notamment.

De nombreuses applications scientifiques, telle que l'imagerie médicale construisent et utilisent des données numériques sous la forme d'une grille à trois dimensions qui sont des grands tableaux tridimensionnels de données. Ces données s'appellent des données volumiques qui sont produites par une multitude d'appareils comme l'imagerie par résonance magnétique (IRM), scanner ... .A noter que ces appareils offrent ce type de données via un ensemble d'images représentant une série des coupes 2D d'un objet. L'organisation des données selon une grille régulière facilite d'une manière importante l'ensemble des traitements réalisés. Pour cela on utilise les techniques de rendu volumique

Le rendu volumique est un outil puissant qui permet de visualiser des données volumiques, c'est une branche de la visualisation scientifique. Le rendu volumique a plusieurs applications importantes tels que : la médecine, la mécanique et l'industrie des films etc... . Il existe trois principales méthodes qui permettent de visualiser des données volumiques : visualisation de tranches 2D, visualisation d'isosurfaces et rendu volumique direct. Dans notre présent travail on est intéressés à la méthode de rendu volumique direct (RVD) par raycasting, appliquée aux images médicales.

Le RVD par le raycasting évalue la couleur de chaque pixel de l'image finale en traçant un rayon à travers le volume à partir de la position de l'observateur. Si le rayon frappe le volume, la couleur du pixel est calculée en échantillonnant les valeurs de données le long du rayon en un nombre fini de positions dans le volume et en les combinant ensemble. La projection d'une séquence de points 3D sur un écran 2D pour déterminer la couleur de chaque pixel est considérée comme une cause principale de l'ambiguïté des images obtenues par toutes les méthodes de l'informatique graphique mais beaucoup plus par le RVD vu la transparence des

données. Les valeurs de transparence correspondant aux données initiales ainsi leurs couleurs sont définies en spécifiant une fonction de transfert appropriée qui présente un déterminant principal de la perception de données dans le rendu final. Donc, son objectif est d'attribuer des propriétés optiques aux données volumiques où la transparence ou l'opacité présente l'absorption tandis que la couleur présente l'intensité.

Dans ce travail on veut appliquer une méthode de raycasting adaptatif qui a le but d'améliorer la qualité de rendu en réduire le nombre de points d'échantillonnage pour rendre de grands ensembles de données où chaque évaluation pourrait nécessiter une décomposition coûteuse. Et aussi augmenter La fréquence d'échantillonnage dans la région d'image où les variations de couleur dans l'espace sont élevées. Mais elle est diminuée dans les régions homogènes. Avec l'augmentation des résolutions d'écran, des images de haute résolution sont créés plus efficacement,

Le présent mémoire est structuré en trois chapitres de la manière suivante :

Dans le premier chapitre nous présenterons de manière générale les notions fondamentales pour le rendu volumique, et nous orienterons notre travail pour mettre l'accent sur le rendu volumique direct par raycasting.

Dans le second chapitre nous présenterons quelques méthodes de raycasting adaptatif, leurs fonctionnements, les recherches et les travaux qui ont déjà fait.

Le troisième chapitre, est divisé en deux parties : la première consacré la présentation de l'architecture globale et détaillé de notre méthode, la deuxième se fait sur la précision de l'environnement de développement matériel et logiciel utilisé pour mettre en œuvre notre application et l'analyse des résultats obtenus.

Ce mémoire est terminé par une conclusion générale récapitulant notre système.



## Références

- [1] Luc Buatois. Algorithmes sur GPU de visualisation et de calcul pour des maillages non-structurés. Modélisation et simulation. Institut National Polytechnique de Lorraine - INPL, 2008. Français.
- [2] [http://nptel.ac.in/courses/Webcourse-contents/IIT-ANPUR/FLUIDMECHANICS/lecture-6/6-3\\_scaler\\_vector\\_field.htm](http://nptel.ac.in/courses/Webcourse-contents/IIT-ANPUR/FLUIDMECHANICS/lecture-6/6-3_scaler_vector_field.htm): des cours fournis par Dr. G.Biswasde le département de génie mécanique Institut indien de technologie de Kanpur et par Dr S.K. Somde la département de génie mécanique Institut indien de technologie de Kharagpur.
- [3] A .Kaufman. « Voxels as a Computational Representation of Geometry». In The Computational Representation of Geometry, Course, ACM SIGGRAPH. 1994.
- [4] A .Smith. « A Pixel Is Not A Little Square, A Pixel Is Not A Little Square, A Pixel Is Not A Little Square! (And a Voxel is Not a Little Cube) ». Technical Report Microsoft, Inc. 1995.
- [5] Alexandre ANCEL, 20 juin 2008, Visualisation volumique de données gyrocinétiques. Mémoire.
- [6] B. Mora. Nouveaux algorithmes interactifs pour la visualisation de données volumiques. PhD thesis, Université ToulouseIII-Paul Sabatier, 07 Décembre 2001.
- [7] <http://cumbia.visus.uni-stuttgart.de/ger/research/proj/ito/materials/> : des cours fournis par Thomas Ertl, Thomas Klein, Katrin Bidmon et Andreas Hub de l'institut de visualisation et systèmes interactif (VIS), Université de Stuttgart. 2003.
- [8] Kenneth Moreland. Fast High Accuracy Volume Rendering. PhD thesis, University of New Mexico, July 2004.
- [9] Nelson Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics, 1(2):99\_108, 1995.
- [10] J. P. Schulze, M. Kraus, U. Lang, and T. Ertl. Integrating pre-integration into the shear-warp algorithm. In VG '03 : Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics, pages 109\_118, New York, NY, USA, 2003. ACM.
- [11] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. Local maximum intensity projection (LMIP) : A new rendering method for vascular visualization. Journal of Computer Assisted Tomography (JCAT).1998.
- [12] Soyer P, Heath D, Bluemke DA, et al. « Three-dimensional helical CT of intrahepatic venous structures: comparison of three rendering techniques».Article de journal : Computer Assisted Tomography, volume 20, numéro 1. 1996.

- [13] Stephane Marchesin, Jean-Michel Dischler, and Catherine Mongenet. Feature Enhancement using Locally Adaptive Volume Rendering. In Hege et al. [HMMS07].
- [14] B. Mora, J.P. Pessel, and R. Caube. A new object-order ray-casting algorithm. In IEEE Visualization, Boston, USA, October 2002.
- [15] Markus Hadwiger; Patric Ljung; Christof Rezk Salama and Timo Ropinski (2009). «Advanced Illumination Techniques for GPU-Based Volume Raycasting».
- [16] Frida Hernell, Patric Ljung and Anders Ynnerman. Local Ambient Occlusion in Direct Volume Rendering.
- [17] Stefan Suwelack, Eric Heitz, Roland Unterhinninghofen and Rüdiger Dillmann .Adaptive GPU Ray Casting Based on Spectral Analysis.
- [18] M. Mustra, K. Delac, and M. Grgic, (2008), ‘Overview of the DICOM standard’ in *ELMAR:50th International Symposium*, IEEE,

## Table des figures

Figure 1.1 Approche graphique de la notion de champ scalaire. ....	3
Figure 1.2 Approche graphique de la notion de champ vectoriel. ....	4
Figure 1.3 Exemples de maillages structurés en deux dimensions. ....	5
Figure 1.4 Exemple de maillage non structurés en deux dimensions. ....	5
Figure 1.5 Représentation de données volumiques. ....	6
Figure 1.6 Rendu volumique direct (RVD). ....	7
Figure 1.7 Les méthodes Forwards ....	8
Figure 1.8 Les méthodes Backwards. ....	8
Figure 1.9 Le principe de Raycasting. ....	9
Figure 1.10 Classification ....	10
Figure 1.11 Comparaison de qualité d'images entre la pré-classification (Gauche) et la post-classification (Droite). ....	11
Figure 1.12 Schéma d'un rayon de lumière traversant un corps participatif ....	12
Figure 1.13 Approximation de l'intégrale par la somme de Riemann. ....	13
Figure 1.14 MIP d'une angiographie par résonance magnétique. ....	14
Figure 1.15 Composition d'avant en arrière. ....	14
Figure 1.16 Composition d'arrière en avant. ....	15
Figure 2.1. L'activation du raffinement de l'intersection isosurface entraîne une amélioration considérable de la qualité de l'image sans impact notable sur les performances. ....	19
Figure 2.2. L'image de gauche illustre un petit détail du modèle de dragon asiatique avec un taux d'échantillonnage de 0,5. À droite, l'échantillonnage adaptatif augmente le taux d'échantillonnage à 4,0 proches de l'iso-surfaces. N ....	19
Figure 2.3 : Division en blocs hiérarchique avec sous-échantillonnage. ....	22
Figure 2.4. Division en blocs hiérarchique avec sous-échantillonnage moyen. ....	23
Figure 2.5. Illustration des voisins de bloc en 2D. ....	24
Figure 2.6. Comparaison des méthodes d'interpolation ....	26
Figure 2.7. comparaison entre l'échantillonnage adaptatif et l'échantillonnage fixé. ....	32
Figure 2.8. Illustration détaillée des distances d'échantillon le long d'un rayon LAO. ....	34
Figure 2.9. Un signal sinusoïdal (noir) est reconstruit avec une méthode de lancer de rayons (rouge). ....	36

Figure 2.10. Une fonction de transfert typique d'opacité non stationnaire utilisé pour la visualisation médicale (gauche). La fonction de transfert est décalée le long de l'axe des x pour parcourir les différentes structures anatomiques d'une main humaine (milieu, droite) .....	38
Figure 3.1 Architecture global .....	43
Figure 3.2. L'architecture détaillée du Ray Casting statique .....	45
Figure 3.3. L'architecture détaillée du Ray Casting adaptatif.....	48
Figure 3.4. Raycasting des pieds, temps de rendu 1.32, distance d'échantionnage: 5 .....	55
Figure 3.5. Raycasting des pieds, temps de rendu 7.14, distance d'échantionnage: 5 .....	56
Figure 3.6. Raycasting des pieds ,temps de rendu 6.41 ,distance d'échantionnage: 5 .....	57
Figure 3.7. Raycasting, temps de rendu 0.33, distance d'échantionnage: 5 .....	58
Figure 3.8. Raycasting, temps de rendu 4.01, distance d'échantionnage: 5 .....	59
Figure 3.9. Raycasting des pieds, temps de rendu 10.61, distance d'échantionnage: 5 .....	60
Figure 3.10.Comparaison entre Visualisation volumique par le Raycasting (gauche) et visualisation traditionnelle des tranches 2D .....	61

# Table des matières

Table des matières

Table des figures

Introduction générale : ..... 1

## Chapitre 1 : l'état de l'art

1. Introduction .....	3
2. Données volumiques .....	3
3. La visualisation scientifique.....	5
4. Méthodes de visualisation volumique .....	6
4.1. Visualisation de tranches 2D .....	6
4.2. Visualisation d'isosurfaces.....	6
4.3. Rendu volumique direct (DVR pour Direct Volume Rendering).....	7
5. Rendu volumique direct .....	7
5.1. Principe.....	7
5.2. Ray Casting.....	8
6. Fonction de transfert et classification.....	9
7.Équation de rendu volumique .....	11
8. Les techniques de Raycasting .....	13
8.1. Maximum Intensity Projection .....	13
8.2. Par accumulation (composition) .....	14
8.2.1. Composition d'avant en arrière.....	14
8.2.2. Composition d'arrière en avant.....	15
9. Conclusion.....	16

## Chapitre 2 : Raycasting adaptatif

1. Introduction .....	17
-----------------------	----

2. Raycasting adaptatif .....	17
2.1 Définition.....	17
3.Échantillonnage adaptatif:.....	17
3.1. Raffinement des intersections.....	19
4- Multirésolution basée Niveau de détail (LOD) et l'échantillonnage adaptatif.....	20
4.1. Représentation multirésolution basée sur octree : .....	20
4.2. Blocs et structures d'accélération.....	21
4.3. Représentations multirésolution hiérarchiques.....	22
5.Échantillonnage de volumes reconstruits par la multirésolution .....	23
5.1.Échantillonnage de bloc le plus proche .....	24
5.2. Échantillonnage d'interpolation interbloc .....	25
5.3. Résultats interbloc d'interpolations .....	27
6. Raycasting sur le GPU .....	28
6.1. Échantillonnage adaptatif dans l'espace objet .....	28
7. Implémentation de l'Occlusion ambiante locale :.....	29
7.1. Traitement multirésolution .....	30
7.2. Pipeline LAO.....	30
7.3.Échantillonnage d'occlusion locale.....	33
8. Application de la théorie de l'échantillonnage de Raycasting.....	35
9. Analyse spectrale.....	37
9.2. Analyse spectrale de la fonction composite .....	37
9.2. Fonctions de transfert dynamique.....	38
9.3. Implémentation.....	39
10. Conclusion.....	41
 <b>Chapitre 3 : Conception et implémentation</b>	
1. Introduction .....	42
2. Motivation et Objectif .....	42

3. Conception .....	43
3.1. Conception globale .....	43
3.1.1. Reconstruction .....	44
3.1.2. Fonction de transfert et classification .....	44
3.1.3. Rendu (Raycasting adaptatif).....	44
3.1.4. Affichage (visualisation de volume).....	44
4. Conception détaillée .....	45
4.1. Introduction .....	45
4.2. Raycasting statique :.....	45
4.3. Raycasting adaptatif .....	48
5. Implémentation.....	53
5.1 Environnement de l'application.....	53
5.1.1. Environnement de développement matériel.....	53
5.1.2. Environnement de développement logiciel.....	53
5.2. Le type d'images médicales traité .....	54
5.3 Tests et Analyse de résultats.....	54
6. Conclusion.....	61
Conclusion générale .....	62