

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M.Ing.

PAR
SÉBASTIEN PICCAND

UTILISATION DE LA NOTION DE RÉGION D'INTÉRÊT POUR LA
VISUALISATION TRIDIMENSIONNELLE D'IMAGES MÉDICALES

MONTRÉAL, LE 10 AOÛT 2005

© droits réservés de Sébastien Piccand

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Mme Rita Noumeir, directrice de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Eric Paquette, codirecteur
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Stéphane Coulombe, président du jury
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Frédéric Lesage, membre du jury
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 27 JUIN 2005

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UTILISATION DE LA NOTION DE RÉGION D'INTÉRÊT POUR LA VISUALISATION TRIDIMENSIONNELLE D'IMAGES MÉDICALES

Sébastien Piccand

SOMMAIRE ABRÉGÉ

Un des problèmes de la visualisation d'images médicales est de pouvoir manipuler en temps réel des données tridimensionnelles en conservant le maximum d'information. Or, l'avancement des techniques d'acquisition permet d'obtenir des données de plus haute définition et les méthodes de visualisations ne permettent plus cette interactivité sans sacrifier de l'information.

Si certaines zones, appelées régions d'intérêts, sont indispensables à un bon diagnostic, les autres régions, appelées contexte, ne servent qu'à bien situer ces régions d'intérêts. Ce mémoire présente une méthode de rendu exploitant cette notion afin de manipuler les données en temps réel et conserver un maximum d'information sur la région d'intérêt.

La méthode utilise une approche multi-résolution à base d'ondelettes en affichant une région d'intérêt en pleine résolution et en sacrifiant la résolution du contexte. L'utilisateur peut alors définir une zone d'intérêt de son choix tout en manipulant les données. L'accélération obtenue atteint un facteur 30 sur le temps de rendu du volume complet et permet alors une navigation interactive.

UTILISATION DE LA NOTION DE RÉGION D'INTÉRÊT POUR LA VISUALISATION TRIDIMENSIONNELLE D'IMAGES MÉDICALES

Sébastien Piccand

SOMMAIRE

Un des problèmes de la visualisation d'images médicales est de pouvoir manipuler en temps réel des données tridimensionnelles en conservant le maximum d'information. Or, grâce à l'avancement des techniques d'acquisition, ces données sont de plus haute définition et nécessitent plus de mémoire pour les stocker et les traiter. Des techniques de visualisation innovantes sont alors nécessaires pour naviguer de façon efficace dans les données. Une méthode couramment utilisée est d'employer une représentation multi-résolution des données afin de pouvoir manipuler les données en basse résolution et de pouvoir obtenir une haute résolution lorsque l'on fixe le point d'observation.

Or, toutes les régions des données n'ont pas la même importance. En effet certaines zones sont indispensables à un bon diagnostic et sont appelées régions d'intérêt. Les autres régions, appelées contexte, sont toutefois nécessaires pour permettre de bien situer ces régions d'intérêt.

Ce mémoire présente une méthode de rendu exploitant cette notion de région d'intérêt afin de satisfaire aux deux exigences principales : manipuler les données en temps réel et conserver un maximum d'information sur la région d'intérêt. La méthode utilise une approche multi-résolution à base d'ondelettes en affichant une région d'intérêt en pleine résolution et en sacrifiant la résolution du contexte. Les données sont d'abord décomposées par une transformée en ondelettes dans une première phase. Le rendu proprement dit utilise la technique de projection d'empreintes d'ondelettes afin d'éviter la reconstruction du volume. L'utilisateur peut alors définir une zone d'intérêt de son choix tout en manipulant les données. Pour améliorer la qualité de ce rendu, les calculs d'illumination sont également appliqués lors d'une phase de pré-calculs. Nous avons en outre proposé une façon de définir et manipuler la zone d'intérêt en utilisant une représentation des données sous forme d'arbres.

L'accélération apportée atteint un facteur 30 par rapport au temps de rendu du volume complet et permet alors une manipulation interactive des données, tout en conservant le maximum d'information sur la région d'intérêt.

VOLUME RENDERING OF MEDICAL IMAGES USING A REGION OF INTEREST

Sébastien Piccand

ABSTRACT

A challenge in visualization of medical images is being able to move in real time a volume of medical images while keeping as much information as possible. The new imaging modalities enable to get more precise data and thus more memory is needed to handle and store it. Innovative image visualization techniques are therefore necessary to be able to move through the data in an interactive way. Using a multiresolution representation of the data enables to move the low resolution representation of the data and to get access to the full resolution when stopping.

However only some structures of interest inside the whole data are very important. These structures typically occupy a small percentage of the data but their analysis requires context information like locations within a specific organ or adjacency to sensitive structures.

This thesis proposes a 3D visualization technique using the notion of region of interest to keep full resolution on the region of interest while achieving efficiency in rendering during the exploratory navigation phase. This technique is based on the multiresolution property of the wavelet transform in order to display a full resolution region of interest while displaying a coarser context. In a preprocessing stage the data is decomposed with a three-dimensional wavelet transform. The interactive visualization process then uses the wavelet representation and a user-specified region to render a full resolution region of interest and a coarser context directly from the wavelet space through wavelet splatting, thus avoiding volume reconstruction. This efficient rendering approach is combined with lighting calculations, in the preprocessing stage to enhance the quality of the rendering. A specific structure based on a zero-tree is also used to define and manipulate the region of interest more easily.

The speed up achieved reaches a factor of 30 against the time needed to display the full resolution data and enables an interactive manipulation of the data, while keeping the maximum information on the region of interest.

REMERCIEMENTS

Je tiens à remercier ma directrice de mémoire, Rita Noumeir, ainsi que mon codirecteur, Eric Paquette, pour la confiance qu'ils m'ont accordée et leur soutien.

La réalisation de ce mémoire a été rendue possible grâce au protocole d'entente entre l'ENST Bretagne et l'ETS. Je remercie tout ceux qui ont contribué à cette entente.

Ce projet a été financé par les subventions à la découverte du CRSNG de Mme Noumeir et de M. Paquette.

Je remercie également tous les membres du LIVIA pour l'ambiance de travail mais aussi pour les activités sociales que j'ai partagées avec eux.

TABLE DES MATIÈRES

	Page
SOMMAIRE.....	i
ABSTRACT.....	ii
REMERCIEMENTS.....	iii
TABLE DES MATIÈRES	iv
LISTE DES TABLEAUX.....	vii
LISTE DES FIGURES	viii
LISTE DES ALGORITHMES	xii
LISTE DES ABRÉVIATIONS ET SIGLES	xiii
INTRODUCTION.....	1
CHAPITRE 1 LES ONDELETTES	8
1.1 Un peu d'histoire	8
1.1.1 De la transformée de Fourier à la transformée de Fourier à fenêtre.....	8
1.1.2 De la transformée de Fourier à fenêtre à la transformée en ondelettes	9
1.1.3 Développement de la transformée en ondelettes.....	10
1.2 La transformée de Fourier à fenêtre	11
1.3 La transformée en ondelettes	14
1.3.1 Transformée en ondelette discrète.....	16
1.4 L'analyse multirésolution	18
1.5 Génération des bancs de filtre	21
1.5.1 Relations entre échelles.....	21
1.6 Propriétés des filtres obtenus	24
1.7 Quelques familles d'ondelettes	24
1.7.1 Ondelette de Haar	24
1.7.2 Autres ondelettes	25
1.7.2.1 Ondelette de Shannon	25
1.7.2.2 Ondelettes de Daubechies	25
1.7.2.3 Ondelettes biorthogonales.....	26
1.7.2.4 Ondelettes de Battle-Lemarié	26

1.7.3	Ondelettes générées par la méthode du lissage.....	27
1.8	Cas multidimensionnel.....	28
1.9	Résumé	31
CHAPITRE 2 ÉTAT DE L'ART		32
2.1	Modèles de rendu	32
2.1.1	Rendu de volume direct.....	33
2.1.2	Rendu radiographique.....	34
2.1.3	Projection du maximum d'intensité.....	35
2.2	Techniques de rendu de volume	35
2.2.1	Lancé de rayon.....	36
2.2.2	Projection de voxels sur le plan de projection.....	37
2.2.3	Méthode <i>Shear-Warp</i>	38
2.2.4	Utilisation du matériel graphique.....	39
2.2.4.1	Textures 2D.....	40
2.2.4.2	Textures 3D.....	40
2.2.5	Techniques à base d'ondelettes	41
2.2.6	Techniques hybrides.....	42
2.3	Représentation du volume avec volume d'intérêt	43
2.3.1	Codage par blocs	44
2.3.2	Codage par arbres	45
2.4	Travaux proches	46
2.5	Résumé	47
CHAPITRE 3 MÉTHODE PROPOSÉE.....		48
3.1	Calculs préliminaires pour le rendu.....	48
3.1.1	Illumination.....	48
3.1.2	Classification	50
3.1.3	Ombrage.....	50
3.2	Projection des empreintes d'ondelettes	51
3.2.1	Avantages de la technique.....	53
3.2.2	Choix de l'ondelette	53
3.2.3	Projection utilisée	54
3.2.4	Génération des empreintes d'ondelettes	54
3.2.4.1	Méthode d'origine	55
3.2.4.2	Modifications.....	58
3.2.5	Projection des empreintes	59
3.2.5.1	Matrice de passage.....	59
3.2.5.2	Influence de l'interpolation	61
3.3	Région d'intérêt.....	61
3.3.1	Région d'intérêt dans le volume transformé.....	64

3.3.2	Structure d'arbres	65
3.4	Résumé	67
CHAPITRE 4 RÉSULTATS ET DISCUSSIONS		68
4.1	Données utilisées	68
4.2	Interface utilisateur	70
4.2.1	Chargement des données	70
4.2.2	Modification des modes de rendu	71
4.2.3	Sélection de la région d'intérêt.....	72
4.2.4	Visualisation des données	72
4.3	Influence de la fonction d'opacité sur le rendu final	73
4.4	Utilisation d'une post-convolution	75
4.5	Apport de la structure hiérarchique des coefficients	76
4.6	Influence de la taille de la région d'intérêt	76
4.7	Résumé	81
CONCLUSION		82
ANNEXE		
1 : Implémentation		85
BIBLIOGRAPHIE		95

LISTE DES TABLEAUX

	Page
Tableau I	Comparaison du temps de génération des empreintes. Le temps affiché est une moyenne sur 22 temps de projections d'empreintes d'ondelettes de Haar, selon des angles différents, pour une décomposition de niveau 3, auxquels la plus haute et la plus basse valeur ont été enlevées..... 59
Tableau II	Comparaison du temps de rendu en basse résolution avec post-convolution ou accumulation d'empreintes 75
Tableau III	Effet de la représentation sous forme d'arbres tronqués du volume. Pour le volume Corps, la fonction d'opacité est celle utilisée à la figure 43(b). Pour le volume Tête, la fonction d'opacité est uniforme..... 77
Tableau IV	Accélération apportée par l'utilisation d'une région d'intérêt. Le temps de rendu est exprimé en secondes. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer le rendu courant et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille du volume total, exprimé en pourcentage 80

LISTE DES FIGURES

	Page
Figure 1	Représentation initiale des données 2
Figure 2	Visualisation des données..... 3
Figure 3	Visualisation des données avec région d'intérêt 5
Figure 4	Un signal complexe et sa DFT..... 11
Figure 5	Le signal original fenêtré à l'instant $t = 500$, et sa DFT 12
Figure 6	Le signal original fenêtré à l'instant $t = 3500$, et sa DFT..... 13
Figure 7	La transformée de Fourier avec deux fenêtres de tailles différentes 13
Figure 8	Le compromis temps-fréquence. À gauche le cas de la transformée de Fourier, au centre le cas du signal original, à droite le cas de la transformée de Fourier à fenêtre 15
Figure 9	Ondelette de Morlet 17
Figure 10	Le compromis temps-fréquence dans le cas de la transformée en ondelettes 18
Figure 11	Réalisation en banc de filtres pour la décomposition 23
Figure 12	Décomposition classique en ondelettes 23
Figure 13	Décomposition en paquets d'ondelettes..... 23
Figure 14	Ondelette de Haar 25
Figure 15	Ondelette de Daubechies D2 26
Figure 16	Ondelette biorthogonale 2.4 27
Figure 17	Représentation de la décomposition d'une image 29
Figure 18	Représentation d'un volume décomposé..... 30

Figure 19	Projection des données	34
Figure 20	Données composées de 1024 tranches de 512×512 pixels rendues selon le même point de vue avec un rendu différent	36
Figure 21	Résumé de la méthode <i>Shear-Warp</i>	39
Figure 22	Les trois piles de textures nécessaires	40
Figure 23	Génération des polygones dans l'axe d'observation	41
Figure 24	Approximation du gradient par deux méthodes	49
Figure 25	Amélioration de la qualité du rendu avec des calculs d'illumination et d'ombrage	51
Figure 26	Angle de vue lors de la visualisation des données	55
Figure 27	Théorème de la tranche de Fourier. Le volume original [1] subit une FFT pour obtenir le volume transformé [2]. La tranche [3] du volume transformé subit une IFFT pour obtenir la projection [4] du volume original	56
Figure 28	Les 8 projections de l'ondelette de Haar. Chaque projection correspond à un type d'ondelette $\{f_{o,i,0}\}_{i \in [0..7]}$	56
Figure 29	Adoucissement des contours de la projection de l'ondelette de Haar en utilisant le théorème de la tranche de Fourier	57
Figure 30	Artefacts de projection en utilisant le théorème de la tranche de Fourier avec la transformée de Fourier discrète de la fonction discrète d'ondelette	58
Figure 31	Système de coordonnées. O est l'origine des deux repères. Le repère de l'espace de visualisation est $(O, \vec{u}, \vec{v}, \vec{n})$, celui des données est $(O, \vec{x}, \vec{y}, \vec{z})$	60
Figure 32	Comparaison du rendu obtenu en utilisant ou non une interpolation bilinéaire sur le plan d'observation	63
Figure 33	Comparaison du rendu obtenu en affichant ou non les données autour de la région d'intérêt	63
Figure 34	Transformation de la région d'intérêt dans l'espace de visualisation	64

Figure 35	Relation spatiale entre les coefficients. Quand le volume est décomposé deux fois, pour reconstruire une région de $2 \times 2 \times 2$ voxels, il faut un coefficient de basse résolution c_0 , 7 coefficients de détails d'ordre 1 $\{d_{1,i,0}\}_{i \in [1..7]}$, et 7×8 coefficients de détails d'ordre 2 $\{d_{2,i,j}\}_{i \in [1..7], j \in [0..7]}$. Les coefficients $d_{1,7,0}$ et $\{d_{2,7,j}\}_{j \in [0..7]}$, non affichés, sont situés dans les sous volumes non visibles	65
Figure 36	Structure obtenue en utilisant une représentation hiérarchique des coefficients	66
Figure 37	Enregistrement des données de l'arbre de façon linéaire	66
Figure 38	Volume Corps	69
Figure 39	Volume Tête	70
Figure 40	Fenêtre de commande	71
Figure 41	Changement du mode de rendu	72
Figure 42	Manipulation du volume	73
Figure 43	Effet de l'opacité sur le rendu. A gauche les fonctions d'opacité et à droite le rendu obtenu	74
Figure 44	Influence de la taille de la région d'intérêt. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer un rendu avec région d'intérêt et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille totale du volume. La fonction d'opacité utilisée est celle de la figure 43(c)	77
Figure 45	Influence de la taille de la région d'intérêt sur la rapidité du rendu lorsque la région est plus petite. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer un rendu avec région d'intérêt et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille totale du volume. La fonction d'opacité utilisée est celle de la figure 43(c)	78
Figure 46	Région d'intérêt centrée sur le foie. La région d'intérêt est de taille $123 \times 123 \times 123$ voxels, soit 1.21% du volume original. Le	

- temps de rendu moyen est de 1.88 seconde, soit un gain en vitesse d'un facteur 10.6 par rapport au rendu pleine résolution 79
- Figure 47 Région d'intérêt centrée sur une partie d'une artère. La région d'intérêt est de taille $52 \times 52 \times 52$ voxels, soit 0.09% du volume original. Le temps de rendu moyen est de 0.65 secondes, soit un gain en vitesse d'un facteur 29.6 par rapport au rendu pleine résolution ... 79
- Figure 48 Région d'intérêt centrée sur un genou. La région d'intérêt est de taille $75 \times 75 \times 75$ voxels, soit 0.27% du volume original. Le temps de rendu moyen est de 0.36 secondes, soit un gain en vitesse d'un facteur 31.1 par rapport au rendu pleine résolution 80
- Figure 49 Diagramme des classes du logiciel développé. Seuls les attributs, méthodes et arguments les plus importants sont représentés 88

LISTE DES ALGORITHMES

	Page
Algorithme 1	51
Algorithme 2	62

LISTE DES ABRÉVIATIONS ET SIGLES

FT	Transformée de Fourier.
DFT	Transformée de Fourier discrète.
WFT	Transformée de Fourier à fenêtre.
WT	Transformée en ondelettes.
QMF	Filtres miroirs en quadrature.
\vec{x}	Vecteur \vec{x} .
\vec{x}^T	Transposée du vecteur \vec{x} .
$\langle \vec{x}, \vec{y} \rangle$	Produit scalaire des vecteurs \vec{x} et \vec{y} .
$\ \vec{x}\ $	Norme du vecteur \vec{x} .
$A * B$	Convolution de A et B .
\emptyset	Ensemble vide.
\cap	Intersection.
\cup	Union.
$A \oplus B$	Espace généré par les espaces indépendants A et B .
$A \perp B$	A et B sont orthogonaux.
<i>text</i>	Texte en anglais.

INTRODUCTION

Les équipements d'acquisition d'images médicales génèrent des données dont le volume occupe une grande place mémoire. Il n'est en effet pas rare qu'un examen de tomographie par résonance magnétique ou par rayon-x génère plus de 500 images pour une occupation mémoire de plusieurs centaines de méga-octets. Le traitement de ces données, pour faciliter la tâche des médecins, est alors de plus en plus complexe. Ainsi, des projets comme TRIP™ (Andriole, 2004) (*Transforming the Radiological Interpretation Process*) de la société SCAR (*The Society for Computer Applications in Radiology*) ont vu le jour afin de résoudre ce problème de gestion des données en soutenant les travaux de recherche .

Représentation des données originales

Les données à traiter sont des images issues de scanners ou de techniques d'imagerie par résonance magnétique. Ces images sont des coupes transversales qui nous donnent de l'information sur la densité des différents tissus traversés. Les données sont donc représentées sous forme d'une pile d'images de densité (figure 1). Les techniques actuelles permettent d'avoir une définition de l'ordre du millimètre au sein de chaque image ainsi qu'entre chaque image.

Un volume dont les voxels sont les pixels de chaque image peut ainsi être défini. Soient H la hauteur des images, L leur largeur et N leur nombre, le volume défini est alors de taille $H \times L \times N$. Dans le cas d'une même résolution entre les pixels selon la hauteur et la largeur de l'image ainsi qu'entre chaque image, le volume est qualifié d'isotrope.

Les objets du volume sont définis comme les parties que l'on veut dissocier à l'intérieur de ce volume. Par exemple un organe et un os sont des objets. Pour différencier ces objets, il faut intervenir manuellement. En effet bien qu'il y ait une relation entre la densité et les types de tissus, cette information n'est pas suffisante car à une même densité peuvent cor-

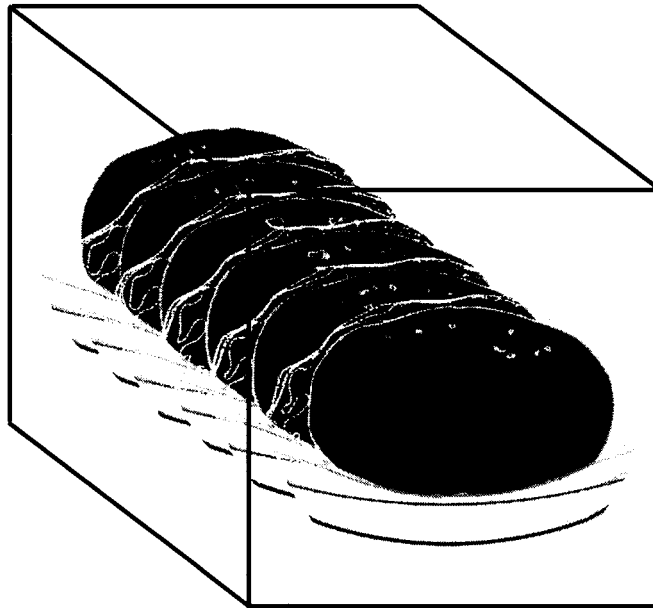


Figure 1 Représentation initiale des données

responder plusieurs tissus. Cette opération, appelée segmentation, est un problème complexe et n'est pas étudié dans le cadre de ce document.

Visualisation des données

Pour visualiser ces données, plusieurs méthodes sont utilisées. Il peut s'agir simplement d'observer les différentes tranches une par une. Il est alors difficile d'interpréter les images puisque l'information tridimensionnelle n'est pas représentée sous une forme naturelle. Des techniques de visualisation volumétrique permettent de palier à ce problème.

Les techniques de visualisation volumétrique permettent en effet de visualiser les données 3D en une seule image projetée sur l'écran (figure 2). Elles consistent à afficher le volume d'une façon transparente en utilisant toute l'information tridimensionnelle. Elles sont basées sur une approche qui permet de contrôler l'opacité des voxels en les projetant sur l'écran de visualisation. Ces méthodes sont très coûteuses en temps de calcul, particulièrement lorsque le volume est composé de nombreux voxels. De plus, une manipulation en

temps réel peut être nécessaire pour mieux analyser l'information tridimensionnelle et le temps de calcul devient alors un problème critique.



Figure 2 Visualisation des données

Les méthodes de rendu de base sont le lancé de rayons (Kajiya et Herzen, 1984 ; Levoy, 1988) ou encore la projection de voxels (Westover, 1990). Dans le cas du lancé de rayon, une intégration s'effectue depuis le plan d'observation et traverse les données, tandis que dans le cas de la projection de voxels, cette intégration se fait à partir des données en direction du plan d'observation. Un compromis profitant des avantages des deux méthodes est la méthode *Shear-Warp* (Lacroute et Levoy, 1994). Elle est connue pour être la méthode logicielle la plus rapide. Les méthodes exploitant le support des textures du matériel graphique (Cabral et al., 1994 ; Rezk-Salama et al., 2000) permettent d'atteindre des rendus temps réels sur des volumes dont l'occupation mémoire ne dépasse pas celle du matériel graphique. Toutefois les données sont désormais de telle grandeur (en nombre de voxels)

que ces méthodes ne suffisent pas à effectuer le rendu en temps réel à pleine résolution et sans compression avec perte.

Rendu multirésolution

Une idée pour accélérer le rendu est d'utiliser une représentation multirésolution des données. Une basse résolution peut être utilisée lorsque le point de vue est déplacé et une haute résolution dès que le point de vue est stabilisé (Guthe et Strasser, 2004 ; Lippert et Gross, 1995). Le rendu peut également être effectué à haute résolution sur une région d'intérêt et à basse résolution sur le contexte (Krishnan et al., 2004 ; LaMar et al., 2000). Les ondelettes sont souvent utilisées (Guthe et Strasser, 2004 ; Krishnan et al., 2004 ; Lippert et Gross, 1995) car elles permettent une représentation multirésolution compacte des données. Lorsque les ondelettes sont utilisées, il est nécessaire de reconstruire le volume avant d'utiliser une des méthodes précédente. Lippert (Lippert et Gross, 1995) propose toutefois une méthode, dite de projection d'empreintes d'ondelettes, permettant d'effectuer un rendu directement à partir des coefficients d'ondelette. La reconstruction s'effectue lors du rendu proprement dit. Au lieu de projeter des voxels sur le plan de projection, des empreintes d'ondelettes pondérées, sortes de textures, sont accumulées sur le plan de projection.

Région d'intérêt

En général, l'utilisateur se concentre sur une région d'intérêt quand il interprète des images médicales. Les structures d'intérêt occupent une petite partie des données mais leur analyse nécessite l'information contextuelle comme leur endroit ou la proximité d'autres structures importantes. Isoler une région d'intérêt permet de réduire le temps de calcul mais sacrifie complètement cette information contextuelle. Une méthode capable de combiner une région d'intérêt affichée en résolution complète avec un contexte affiché en faible résolution (figure 3) permet de préserver la valeur diagnostique des données tout en réduisant le temps de calcul.

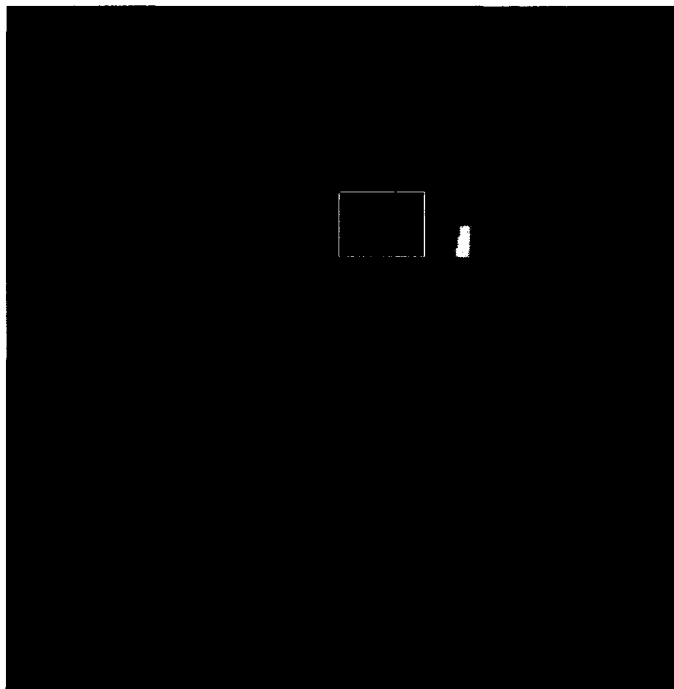


Figure 3 Visualisation des données avec région d'intérêt

La notion de région d'intérêt est très souvent utilisée pour compresser les données originales (Lippert, 1998). Ceci est intéressant lorsque la région d'intérêt est connue d'avance. Lorsqu'elle n'est pas connue d'avance, utiliser une région d'intérêt peut être utile afin d'accélérer le rendu et donc de permettre une navigation plus agréable.

Rendu multirésolution avec région d'intérêt variable

Les méthodes permettant d'effectuer un tel rendu sont celles proposées par Krishnan (Krishnan et al., 2004) et par Lamar (LaMar et al., 2000).

La méthode de Lamar est basée sur un rendu hardware à base de textures. La multirésolution est obtenue en utilisant une structure pyramidale et induit donc un coût en mémoire. L'utilisation de la propriété de multirésolution des ondelettes permettra de réduire cet espace mémoire.

La méthode de Krishnan utilise une représentation des données en adéquation avec la norme JPEG2000. Les données sont décomposées sous forme de blocs qui sont décomposés en ondelettes. La localisation de la région d'intérêt est effectuée en sélectionnant les blocs qui y contribuent. L'utilisation de blocs semble surfaite puisque les ondelettes permettent déjà de sélectionner les coefficients contribuant à une zone d'intérêt en vertu de leur propriété de localisation. D'autre part cette méthode nécessite de reconstruire chaque bloc avant d'effectuer le rendu.

Le but de ce mémoire est d'allier deux propriétés intéressantes des ondelettes, la multirésolution et la localisation, qui ne sont pas exploitées en même temps dans le cadre de la visualisation de données médicales. Il s'agit de montrer qu'il est techniquement possible d'exploiter ces propriétés afin d'effectuer un rendu multirésolution avec région d'intérêt. Il ne couvre pas la validation de la méthode par un utilisateur, se concentrant sur le développement de la méthode de visualisation.

Nous proposons ainsi d'améliorer la méthode de projections d'empreintes d'ondelettes en ajoutant :

- des précalculs d'illumination et ombrage permettant une meilleure interprétation des données ;
- une structure de donnée permettant d'accéder rapidement aux différentes régions des données ;
- une représentation de la région d'intérêt permettant de modifier facilement cette dernière via une interface utilisateur.

Les résultats obtenus sur nos données montrent alors une accélération d'un facteur 30 sur le temps de rendu lorsqu'une petite région d'intérêt est utilisée, ce qui permet une navigation sans avoir à changer de résolution.

Ainsi, après une étude de la théorie des ondelettes pour justifier leur intérêt et une étude des techniques de visualisation existantes, ce mémoire présente une méthode de visualisation

permettant de manipuler des données composées de nombreux voxels en affichant une région d'intérêt, modifiable, en haute résolution, et le contexte en basse résolution. Cette méthode permet ainsi de combiner une bonne représentation des données intéressantes et un temps de calcul court.

CHAPITRE 1

LES ONDELETTES

Ce chapitre a pour but d'expliquer, pourquoi la théorie des ondelettes a été développée et en quoi elle consiste. La transformée de Fourier est supposée connue.

1.1 Un peu d'histoire

L'origine des ondelettes n'est pas très facile à déterminer. En effet elles sont le résultat de la synthèse de plusieurs travaux menés parallèlement dans des domaines variés, tels que la géophysique, la théorie quantique, l'analyse par harmoniques et la vision par ordinateur. Dans chacun de ces domaines, des scientifiques ont eu besoin de développer des outils adaptés à leurs problèmes. Ces outils se sont avérés être très proches des ondelettes. L'origine des ondelettes a été retracée par Ingrid Daubechies dans un rapport à l'université de Princeton (Daubechies, 1995).

1.1.1 De la transformée de Fourier à la transformée de Fourier à fenêtre

En 1807, Fourier propose une méthode de décomposition orthogonale de signaux périodiques. Il s'agit de la série de Fourier, plus tard étendue à la transformée de Fourier (FT pour *Fourier Transform* en anglais).

On notera qu'en 1910, Haar propose une autre base orthonormale pour décomposer les signaux. Le nom d'ondelettes apparaît pour la première fois dans une annexe à la thèse de Haar.

La transformée de Fourier n'est pas toujours adaptée à l'analyse de signaux. En effet, si une haute fréquence apparaît seulement à un moment bref dans un signal, la FT pourra seulement indiquer qu'elle est présente, mais ne pourra pas indiquer le moment auquel elle apparaît. Ainsi en 1946, Gabor propose la transformée de Fourier à fenêtre (WFT pour

Windowed Fourier Transform en anglais). Il s'agit de faire glisser une fenêtre (typiquement une gaussienne) sur le signal avant de lui appliquer une FT. On obtiendra ainsi la FT de la partie du signal située au centre de la fenêtre. Ainsi si une haute fréquence est présente à un moment donné, celle-ci sera détectée uniquement lorsque la fenêtre sera positionnée autour de ce moment ; ce que l'on cherchait à obtenir.

1.1.2 De la transformée de Fourier à fenêtre à la transformée en ondelettes

À la fin des années 1970, le géophysicien Morlet a des besoins plus précis pour analyser des signaux sismiques. Il voudrait une bonne localisation temporelle des hautes fréquences mais également une bonne résolution fréquentielle dans les basses fréquences. Il pense à utiliser une WFT dont on pourrait modifier la largeur de la fenêtre en la contractant ou en la dilatant. C'est ainsi que naît la transformée en ondelettes (WT pour *Wavelet Transform* en anglais).

Par la suite, Morlet a pris contact avec Grossman, un physicien plus théorique. Ils parviennent à construire une formule d'inversion à la transformée de Morlet en 1984. Ce qui est étonnant, c'est que cette formule semble avoir été élaborée plus tôt par Alasken et Klauder dans le domaine de la mécanique quantique. Grossman ayant été le directeur de thèse de Daubechies, c'est naturellement qu'elle est venue à travailler sur le sujet dès 1985.

D'autre part, Meyer, un pure mathématicien, apprend en 1985 les travaux de Morlet et Grossman et y reconnaît une formule introduite par Calderón dans les années 1960 dans le domaine de l'analyse harmonique. Il s'intéresse aux séries d'ondelettes et construit une base orthonormale d'ondelettes. Cette construction est par la suite généralisée à N dimensions avec l'aide de Lemarié. Lemarié à son tour conçoit une base d'ondelettes à partir de fonctions splines. Là encore ce qui est étonnant, c'est qu'une autre base orthonormale d'ondelettes est développée par Stromberg (un mathématicien dans le domaine de l'analyse harmonique) légèrement avant et que Battle (un physicien intéressé par la théorie

quantique) réalise lui aussi une base d'ondelettes à partir de fonctions spline au même moment que Lemarié.

1.1.3 Développement de la transformée en ondelettes

En 1986, Mallat, spécialisé en vision par ordinateur et analyse d'images, reconnaît l'intérêt des bases d'ondelettes pour son domaine. En effet en analyse d'images les structures pyramidales sont utilisées pour analyser une image à différents niveaux de résolution. Il contacte ainsi Meyer et ensemble, ils définissent les principes mathématiques de l'analyse multirésolution. Ils en arrivent également à l'élaboration d'une méthode de calcul rapide de la transformée en ondelettes à l'aide de bancs de filtres. Malheureusement avec les bases connues (Meyer, Battle-Lemarié), les filtres sont infinis et doivent être tronqués. Ainsi en 1987, Daubechies décide de raisonner dans l'autre sens. Les bases d'ondelettes sont générées en partant de filtres. Elle réalise ainsi des bases orthonormales d'ondelettes à support compact.

Or, ces filtres présentent de fortes similarités avec des filtres (QMF pour *Quadrature Mirror Filters* en anglais) utilisés pour la décomposition en sous-bandes en génie électrique. Notamment, Smith et Barnwell en 1983 et Vetterli un peu plus tard, construisent de tels filtres permettant une reconstruction exacte du signal. Ensuite, bénéficiant des approches plus théoriques (issues des mathématiques) et des approches plus pratiques (issues du génie électrique), la conception de bases ondelettes a été améliorée et étendue à d'autres domaines comme l'analyse numérique, l'approximation ou les statistiques.

Ainsi, la transformée en ondelettes a étonnamment été le fruit de plusieurs années de collaboration entre des chercheurs de domaine assez variés. Par la suite, quelques points de cette histoire seront plus détaillés.

1.2 La transformée de Fourier à fenêtre

La transformée de Fourier à fenêtre est utilisée pour répondre au problème de l'analyse temps fréquence. La transformée de Fourier s'avère en effet inefficace lorsqu'un signal bref est inséré dans un signal à évolution temporelle lente. Pour illustrer cela, tel signal est généré et sa transformée de Fourier discrète DFT (pour *Discrete Fourier Transform* en anglais) est calculée (voir la figure 4). Le signal présenté comporte un bref passage

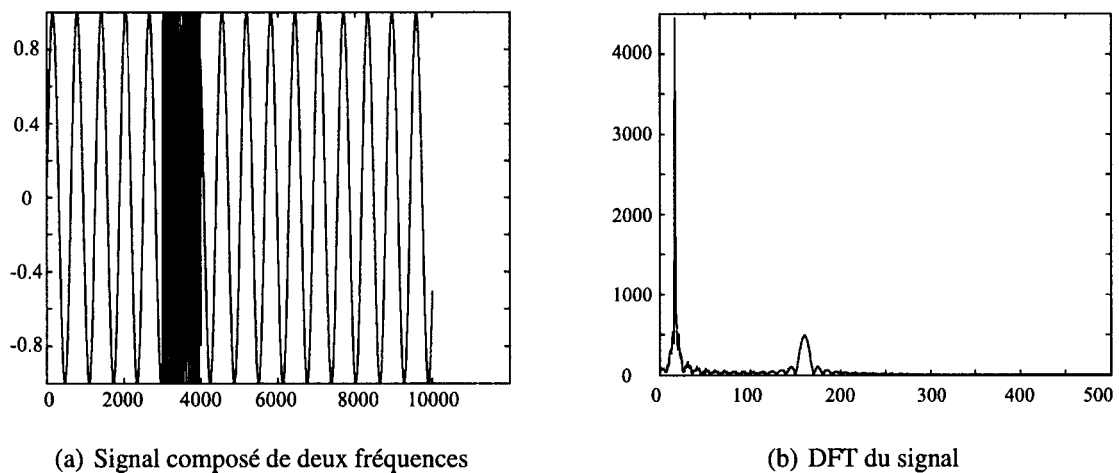


Figure 4 Un signal complexe et sa DFT

lors duquel sa fréquence est multipliée par 10. On voit que la FT permet de détecter cette accélération mais on ne sait pas du tout à quel moment celle-ci a lieu.

Pour répondre à ce problème, une fenêtre d'analyse glissant sur le signal est utilisée. Pour éviter d'introduire des hautes fréquences, cette fenêtre doit être décroissante à l'infini (une fenêtre rectangulaire introduirait des coupures qui provoqueraient des hautes fréquences). D'autre part, d'après le principe d'Heisenberg, plus la fenêtre est large, plus on se rapprochera d'une FT classique avec une bonne résolution fréquentielle mais une mauvaise localisation temporelle. En contre partie, plus cette fenêtre sera étroite plus la résolution temporelle sera bonne mais la résolution fréquentielle sera plus mauvaise. Le compromis idéal est atteint pour une fenêtre gaussienne appelée fenêtre de Gabor.

De façon plus formelle, appelons le signal à étudier dans le domaine temporel $x(t)$ et la fenêtre d'analyse $w(t)$. La fenêtre translatée de τ est donc $w(t - \tau)$. La WFT $X(\tau, f)$ du signal $x(t)$ s'écrit alors :

$$X(\tau, f) = \int_{-\infty}^{+\infty} x(t)w^*(t - \tau)e^{-j2\pi ft} dt. \quad (1.1)$$

En prenant une fenêtre de largeur infinie ($w(t) = 1, \forall t$), la formule de la transformée de Fourier est obtenu. En prenant une fenêtre infiniment étroite (assimilable à un Dirac), on obtient simplement $X(\tau, f) = x(\tau)e^{-j2\pi f\tau}$, c'est à dire, en prenant $f = 0$, le signal original.

Les figures 5 et 6 montrent les résultats de la WFT avec une fenêtre gaussienne de largeur 1000 points, centrée sur $t=500$ (dans la zone de basse fréquence) et $t=3500$ (dans la zone de haute fréquence) pour le signal présenté précédemment. À $t=500$ (figure 5), seule la

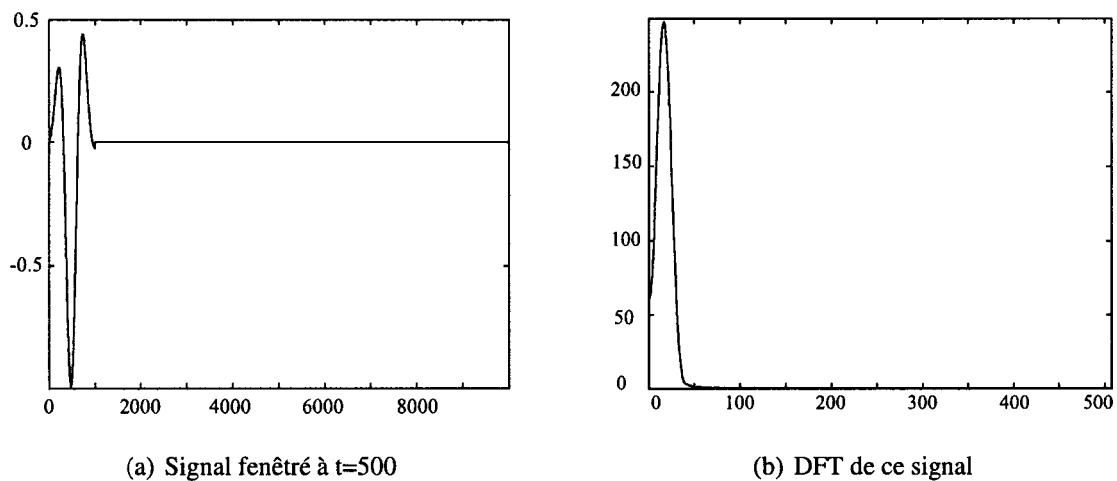


Figure 5 Le signal original fenêtré à l'instant $t = 500$, et sa DFT

basse fréquence est détectée. Ainsi, à cet instant, seule la basse fréquence est présente.

À $t=3500$ (figure 6), seule la haute fréquence est détectée. Ainsi, à cet instant, seule la haute fréquence est présente. La WFT permet donc effectivement d'obtenir une localisation temporelle des fréquences.

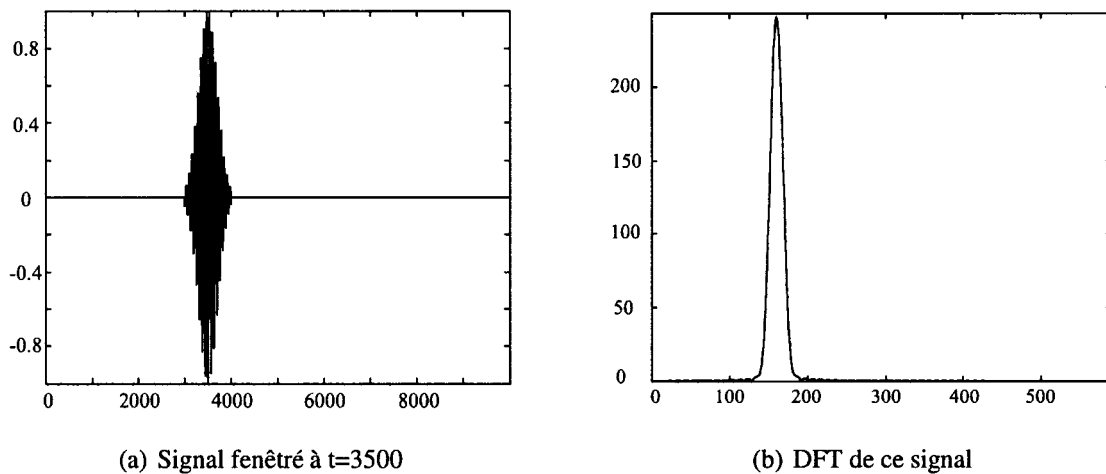


Figure 6 Le signal original fenêtré à l'instant $t = 3500$, et sa DFT

La figure 7 permet d'observer l'influence de la taille de la fenêtre sur la résolution fréquentielle. L'effet sur la résolution temporelle se résume de cette façon : plus la fenêtre est étroite plus on est précis sur le positionnement de l'analyse et donc plus la résolution temporelle est précise). Ainsi, avec une fenêtre 4 fois plus petite le pic correspondant à la

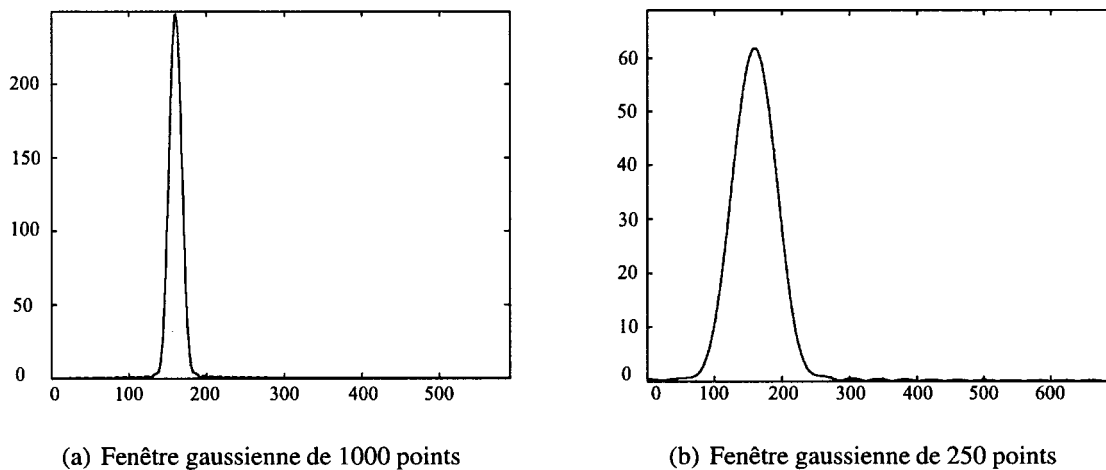


Figure 7 La transformée de Fourier avec deux fenêtres de tailles différentes

fréquence présente dans le signal est vraiment moins net, d'où une résolution fréquentielle plus faible.

Ce compromis entre résolution spatiale ou temporelle (Δt) et résolution fréquentielle (Δf) est justifié par la relation d'incertitude :

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi} \quad (1.2)$$

Le meilleur compromis est obtenu lorsque l'égalité de l'équation 1.2 est vérifiée. La fenêtre gaussienne, dite de Gabor (Gabor, 1946) est alors obtenue :

$$w(t) = \frac{1}{(2\pi)^{\frac{1}{4}} \sqrt{\sigma}} e^{-\frac{t^2}{4\sigma^2}}. \quad (1.3)$$

Comme il s'agit essentiellement de signaux numériques, donc échantillonnés, la représentation de la figure 8 est souvent utilisée pour illustrer la finesse de résolution dans les domaines fréquentiels et temporels. Plus les bandes sont étroites, plus la résolution est fine. Ainsi, pour le signal original, les bandes sont étroites le long de t et longues le long de f car la résolution temporelle est très fine (du niveau de l'échantillon) alors que la résolution fréquentielle est nulle. Dans le cas de la FT, c'est la résolution fréquentielle qui est fine tandis que la résolution temporelle est nulle. Enfin, dans le cas de la WFT, un compromis dans les deux domaines est obtenu.

Il a été montré que la reconstruction parfaite du signal est théoriquement possible en utilisant une autre fenêtre $w'(t)$. On a alors :

$$x(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} X(\tau, f) w'(t - \tau) e^{j2\pi f \tau} d\tau df. \quad (1.4)$$

Les deux fenêtres $w'(t)$ et $w(t)$ doivent alors vérifier $\int_{-\infty}^{+\infty} w'(t) w^*(t) dt = 1$.

1.3 La transformée en ondelettes

On a vu lors de l'historique que la transformée en ondelettes s'est fortement inspirée de la transformée de Fourier à fenêtre. En effet le problème de Morlet était de bien localiser les imperfections du signal qui se traduisent par l'apparition de hautes fréquences. Par

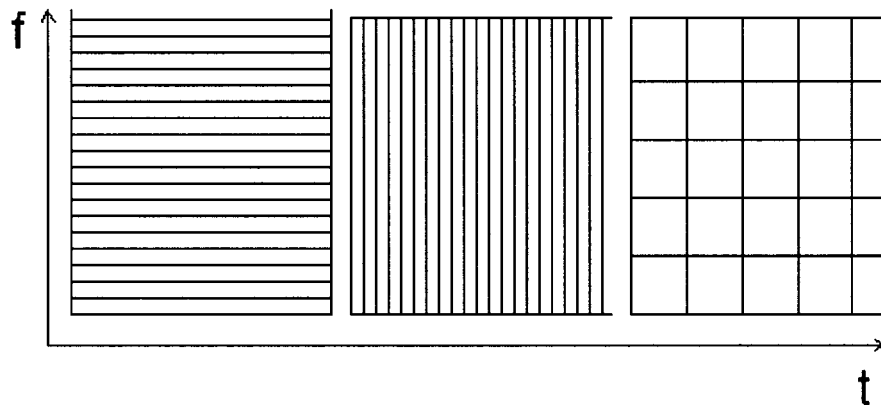


Figure 8 Le compromis temps-fréquence. À gauche le cas de la transformée de Fourier, au centre le cas du signal original, à droite le cas de la transformée de Fourier à fenêtre

ailleurs le comportement global du signal l'intéressait également, ce qui explique pourquoi une bonne résolution dans les basses fréquences lui était nécessaire. Bien sûr il aurait pu utiliser plusieurs WFT avec des tailles de fenêtre différentes. Cependant il aurait obtenu des informations qu'il ne désirait pas et le calcul aurait été relativement long. Ce qu'il lui fallait, ce sont des fenêtres paramétriques qui soient larges dans les basses fréquences et étroites dans les hautes fréquences.

En revenant à la formule de la transformée de Fourier à fenêtre (équation 1.1) :

$$X(\tau, f) = \int_{-\infty}^{+\infty} x(t)w^*(t - \tau)e^{-j2\pi ft} dt. \quad (1.5)$$

En notant $\psi_{\tau, f}(t) = w(t - \tau)e^{j2\pi ft}$, on a :

$$X(\tau, f) = \int_{-\infty}^{+\infty} x(t)\psi_{\tau, f}^*(t) dt. \quad (1.6)$$

Effectuer une « bonne » décomposition du signal revient donc à choisir un bon choix de fonctions $\psi_{\tau, f}(t)$.

C'est ce qui est fait pour la transformée en ondelettes. En partant d'une ondelette dite mère $\psi(t)$, on tire les fonctions :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), b \in \mathbb{R}, a > 0. \quad (1.7)$$

Le paramètre b définit la translation temporelle, tandis que le paramètre a définit le facteur d'échelle qui se traduit par une fenêtre plus ou moins large. La transformée en ondelettes est alors la décomposition du signal sur ces fonctions :

$$C_{a,b} = \int_{-\infty}^{+\infty} x(t)\psi_{a,b}^*(t)dt = \langle x(t), \psi_{a,b}(t) \rangle \quad (1.8)$$

où $\langle \rangle$ désigne un produit scalaire dans l'espace des fonctions de carré intégrable.

Le problème revient donc au choix de cette ondelette mère $\psi(t)$. Pour que celle-ci soit une ondelette, sa transformée de Fourier $\Psi(\omega)$ doit satisfaire la condition de l'équation 1.9 qui assure que la transformée inverse existe.

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty. \quad (1.9)$$

En particulier, cette condition implique que $\Psi(0) = 0$, c'est-à-dire $\int \psi(t)dt = 0$. L'ondelette doit donc être de moyenne nulle. Ceci est donc bien un signal oscillant. Comme il s'agit également d'une fenêtre, elle tend vers 0 à l'infini. C'est sans doute ces deux propriétés qui ont donné à ces fonctions le nom d'ondelettes (petites ondes).

Quant à la transformée inverse, elle est obtenue selon l'équation 1.10.

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C_{a,b}\psi_{a,b}(t) \frac{dad b}{a^2}. \quad (1.10)$$

La première ondelette déterminée par Morlet est une gaussienne modulée. Elle a l'allure de la figure 9.

1.3.1 Transformée en ondelette discrète

En pratique, une analyse du signal pour toutes les valeurs réelles des paramètres d'échelle et de translation ne peut se faire. Il convient donc de les échantillonner pour obtenir une

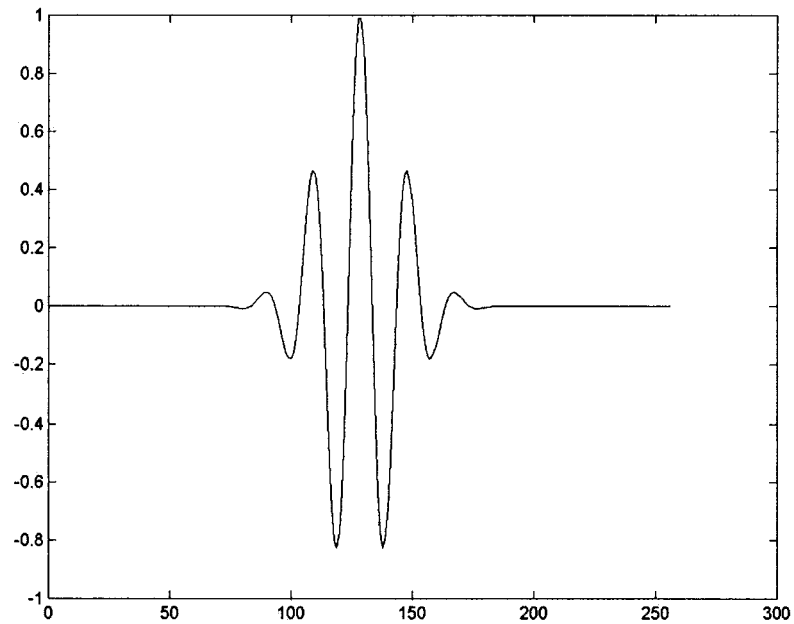


Figure 9 Ondelette de Morlet

grille dans l'espace temps-échelle. On pose ainsi :

$$\begin{aligned} a &= a_0^j, \\ b &= kb_0 a_0^j, \end{aligned} \tag{1.11}$$

avec j et k des entiers. Les fonctions d'ondelettes $\psi_{a,b}(t)$ deviennent donc :

$$\psi_{j,k}(t) = a_0^{-\frac{j}{2}} \psi(a_0^{-j}t - kb_0), \tag{1.12}$$

et on calcule les coefficients d'ondelettes par :

$$c_{j,k} = \langle x(t), \psi_{j,k}(t) \rangle. \tag{1.13}$$

La question est maintenant de savoir si les $(c_{j,k})$ permettent de déterminer complètement le signal original $x(t)$, autrement dit, si l'inversion de la transformée est réalisable. Cela est possible si l'ensemble des fonctions $\psi_{j,k}(t)$ est complet dans l'espace des fonctions de carré intégrable. Dans ce cas l'ensemble constitue une *frame*. Une *frame* n'est par contre pas toujours une base et la décomposition n'est donc pas forcément unique.

Or, dans le cas particulier où $a_0 = 2$ et $b_0 = 1$, les fonctions $\psi_{j,k}(t)$ forment une base orthonormale. Elles permettent de générer n'importe quelle fonction de carré intégrable, ce qui est exactement ce qui était désiré. D'autre part les fonctions sont indépendantes (puisque c'est une base) ce qui garantit l'unicité de la décomposition. Dans ce cas, le changement d'échelle subit un facteur 2 et le compromis temps-fréquence devient celui illustré dans la figure 10.

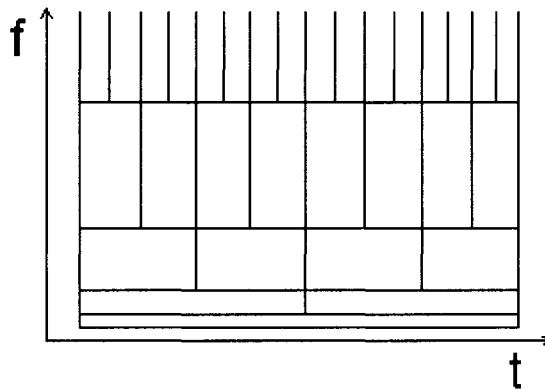


Figure 10 Le compromis temps-fréquence dans le cas de la transformée en ondelettes

1.4 L'analyse multirésolution

En pratique, la décomposition par ondelettes est implémentée à l'aide de bancs de filtres. Cette méthode provient de l'analyse multirésolution proposée par Mallat et Meyer (Mallat, 1989). On se place toujours dans l'espace des fonctions de carré intégrable $L^2(\mathbb{R})$.

Une analyse multirésolution est définie par une séquence de sous-espaces V_i emboîtés tels que :

$$\begin{aligned}
& \forall i \in \mathbb{Z}, V_i \subset V_{i-1}; \\
& \bigcap_{-\infty}^{+\infty} V_i = \emptyset; \\
& \bigcup_{-\infty}^{+\infty} V_i \text{ est dense}^1 \text{ dans } L^2(\mathbb{R}); \\
& x_j(t) \in V_j \Leftrightarrow x_j(2t) \in V_{j-1}; \\
& \exists \phi_{0,0}(t) \text{ appelée fonction d'échelle, telle que} \\
& \quad \{\phi_{0,k}(t) = \phi_{0,0}(t - k), k \in \mathbb{Z}\} \text{ est une base de } V_0.
\end{aligned} \tag{1.14}$$

Une base de V_j est alors l'ensemble $\{\phi_{j,k}(t) = 2^{-\frac{j}{2}}\phi_{0,0}(2^{-j}t - k), k \in \mathbb{Z}\}$. Le signal continu $x(t)$ est alors décomposé à diverses résolutions par décomposition sur ces bases. Ainsi pour une résolution de niveau j , le signal $x_j(t)$ est obtenu par :

$$c_{j,k} = \langle x_j(t), \phi_{j,k}(t) \rangle \tag{1.15}$$

$$x_j(t) = \sum_k c_{j,k} \phi_{j,k}(t). \tag{1.16}$$

Dans le cas d'un signal $x(t)$ discret, le niveau de décomposition 0 est celui prenant en compte tous les échantillons du signal.

L'analyse multirésolution permet d'obtenir une décomposition du signal à chaque niveau de décomposition. Mais il serait intéressant de pouvoir passer d'une résolution à une autre sans avoir à recalculer toute la décomposition. Pour cela on s'intéresse au sous-espace complément W_j de V_j dans V_{j-1} . Plus particulièrement on s'intéresse au sous-espace orthogonal et aux fonctions permettant de générer ce sous-espace.

$$W_j \oplus V_j = V_{j-1} \text{ et } W_j \perp V_j. \tag{1.17}$$

¹ Un ensemble A est dense dans un ensemble B si A est inclus dans B et si chaque élément de B peut être approché indéfiniment (aussi près qu'on veut) par des éléments de A .

Les fonctions permettant de générer ce sous-espace ne sont autres que les ondelettes $\psi_{j,k}(t)$ reliées à l'ondelette mère $\psi_{0,0}(t)$ par la relation :

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi_{0,0}(2^{-j}t - k). \quad (1.18)$$

Ainsi, comme W_j est le complément de V_j dans V_{j-1} , on peut trouver $x_j^\perp(t)$ tel que :

$$x_{j-1}(t) = x_j(t) + x_j^\perp(t). \quad (1.19)$$

D'autre part, $\{\psi_{j,k}(t), k \in \mathbb{Z}\}$ étant une base de W_j , $x_j^\perp(t)$ peut s'écrire sous la forme :

$$d_{j,k} = \langle x_j^\perp(t), \psi_{j,k}(t) \rangle \quad (1.20)$$

$$x_j^\perp(t) = \sum_k d_{j,k} \psi_{j,k}(t). \quad (1.21)$$

Tous les espaces W_j sont mutuellement orthogonaux, les ondelettes le sont donc aussi entre échelle (ce qui n'est pas le cas des fonctions d'échelles) :

$$\langle \psi_{j,k}(t), \psi_{i,l}(t) \rangle = \delta_{i,j} \delta_{k,l}. \quad (1.22)$$

D'après l'équation 1.19, on a :

$$\begin{aligned} \langle x_{j-1}(t), \phi_{j,k}(t) \rangle &= \langle x_j(t), \phi_{j,k}(t) \rangle + \langle x_j^\perp(t), \phi_{j,k}(t) \rangle \\ &= c_{j,k} + 0. \end{aligned} \quad (1.23)$$

Par récurrence, on montre donc que :

$$c_{j,k} = \langle x_0(t), \phi_{j,k}(t) \rangle = \langle x(t), \phi_{j,k}(t) \rangle. \quad (1.24)$$

De la même façon on montre que :

$$d_{j,k} = \langle x_0(t), \psi_{j,k}(t) \rangle = \langle x(t), \psi_{j,k}(t) \rangle. \quad (1.25)$$

Ainsi le signal original provient de la somme de tous les détails des résolutions précédentes :

$$\begin{aligned}
s(t) = s_0(t) &= s_1(t) + \sum_k \langle s(t), \psi_{1,k}(t) \rangle \psi_{1,k}(t) \\
&= s_2(t) + \sum_k \langle s(t), \psi_{2,k}(t) \rangle \psi_{2,k}(t) + \sum_k \langle s(t), \psi_{1,k}(t) \rangle \psi_{1,k}(t) \\
&= \quad \quad \quad \vdots \quad \quad \quad + \quad \quad \quad \vdots \\
&= \sum_j \sum_k \langle s(t), \psi_{j,k}(t) \rangle \psi_{j,k}(t).
\end{aligned} \tag{1.26}$$

1.5 Génération des bancs de filtre

Au lieu de déterminer les coefficients de la décomposition en effectuant directement le produit scalaire des données originales avec les fonctions d'ondelettes, Vetterli (Vetterli et Herley, 1992) propose une méthode rapide de calcul de ces coefficients en utilisant des bancs de filtre.

1.5.1 Relations entre échelles

Comme vu précédemment $V_0 \subset V_{-1}$ et $W_0 \subset V_{-1}$ donc il existe une relation d'échelle telle que :

$$\begin{aligned}
\phi(t) &= \phi_{0,0}(t) = 2 \sum_k h_0(k) \phi_{-1,k}(t) = 2 \sum_k h_0(k) \phi(2t - k), \\
\psi(t) &= \psi_{0,0}(t) = 2 \sum_k h_1(k) \phi_{-1,k}(t) = 2 \sum_k h_1(k) \phi(2t - k).
\end{aligned} \tag{1.27}$$

Or, on a également la relation entre niveaux d'échelle :

$$\sum_{k''} c_{-1,k''} \phi_{-1,k''}(t) = \sum_k c_{0,k} \phi_{0,k}(t) + \sum_{k'} d_{0,k'} \psi_{0,k'}(t). \tag{1.28}$$

En effectuant le produit scalaire avec $\phi_{0,j}(t)$, on obtient :

$$\begin{aligned}
\sum_{k''} c_{-1,k''} \langle \phi_{-1,k''}(t), \phi_{0,j}(t) \rangle &= \sum_k c_{0,k} \langle \phi_{0,k}(t), \phi_{0,j}(t) \rangle \\
&\quad + \sum_{k'} d_{0,k'} \langle \psi_{0,k'}(t), \phi_{0,j}(t) \rangle,
\end{aligned}$$

or

$$\begin{aligned}
\langle \psi_{0,k}(t), \phi_{0,j}(t) \rangle &= 0, \\
\langle \phi_{0,k}(t), \phi_{0,j}(t) \rangle &= \delta_{j,k}, \\
\langle \phi_{-1,k}(t), \phi_{0,j}(t) \rangle &= 2 \sum_n h_0(n) \langle \phi_{-1,k}(t), \phi_{-1,2j+n}(t) \rangle \\
&= 2 \sum_n h_0(n) \frac{\sqrt{2}}{2} \delta_{k,2j+n} \\
&= \sqrt{2} h_0(k - 2j).
\end{aligned}$$

Donc :

$$c_{0,j} = \sqrt{2} \sum_{k''} c_{-1,k''} h_0(k'' - 2j).$$

De même, on obtient :

$$d_{0,j} = \sqrt{2} \sum_{k''} c_{-1,k''} h_1(k'' - 2j).$$

En généralisant aux résolutions j et $j - 1$, on obtient :

$$\begin{aligned}
c_{j,k} &= \sqrt{2} \sum_{k''} c_{j-1,k''} h_0(k'' - 2k), \\
d_{j,k} &= \sqrt{2} \sum_{k''} c_{j-1,k''} h_1(k'' - 2k).
\end{aligned} \tag{1.29}$$

La décomposition n'utilise donc plus la forme explicite des ondelettes mais le rapport interéchelles de ces dernières. La structure permettant de calculer les coefficients à l'échelle j à partir des coefficients à l'échelle $j - 1$ est donc de la forme de la figure 11.

De manière similaire, la relation permettant la reconstruction du signal est donnée par l'équation 1.30.

$$\frac{1}{\sqrt{2}} c_{j-1,k''} = \sum_k c_{j,k} h_0(k'' - 2k) + \sum_k d_{j,k} h_1(k'' - 2k). \tag{1.30}$$

Le filtre h_0 peut être vu comme un filtre passe-bas et h_1 comme un filtre passe haut. La décomposition en ondelettes consiste donc en une succession de filtrages passe-haut et passe-bas à la manière de la figure 12.

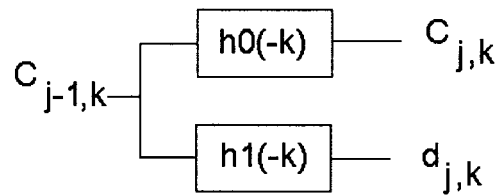


Figure 11 Réalisation en banc de filtres pour la décomposition

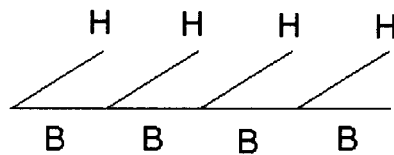


Figure 12 Décomposition classique en ondelettes

Une décomposition en paquets d'ondelettes peut également être effectuée. Ceci revient à poursuivre la décomposition également dans les hautes fréquences (figure 13).

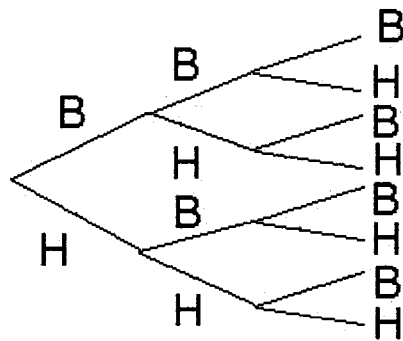


Figure 13 Décomposition en paquets d'ondelettes

1.6 Propriétés des filtres obtenus

On appelle H_0 et H_1 les transformées de Fourier respectives de h_0 et h_1 .

Les filtres sont conjugués en quadrature :

$$\begin{aligned} |H_0(f)|^2 + |H_0(f + \frac{1}{2})|^2 &= 1, \\ |H_1(f)|^2 + |H_1(f + \frac{1}{2})|^2 &= 1. \end{aligned} \tag{1.31}$$

Les ondelettes étant orthogonales, on a la relation suivante :

$$H_0\left(f + \frac{1}{2}\right) H_1^*\left(f + \frac{1}{2}\right) + H_0(f) H_1^*(f) = 0 \tag{1.32}$$

D'autre part :

$$\begin{aligned} H_0(0) = 1 \quad \text{et} \quad \sum_n h_0(n) &= 1, \\ H_1(0) = 0 \quad \text{et} \quad \sum_n h_1(n) &= 0. \end{aligned} \tag{1.33}$$

Enfin les filtres étant orthonormaux :

$$\begin{aligned} \sum_n h_0^2(n) &= 1, \\ \sum_n h_1^2(n) &= 1. \end{aligned} \tag{1.34}$$

1.7 Quelques familles d'ondelettes

1.7.1 Ondelette de Haar

C'est l'ondelette la plus simple. Elle possède une très bonne localisation temporelle et donc une mauvaise localisation fréquentielle.

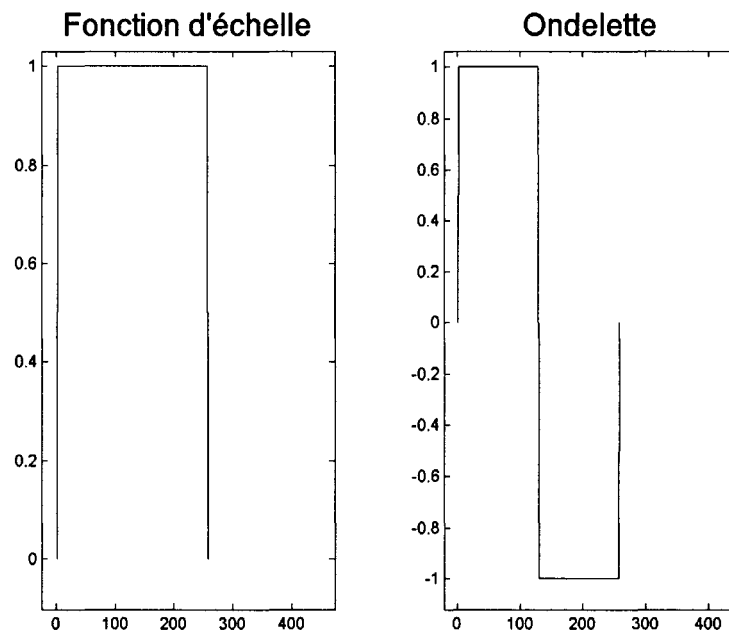


Figure 14 Ondelette de Haar

1.7.2 Autres ondelettes

1.7.2.1 Ondelette de Shannon

Il s'agit de l'ondelette aux propriétés contraires de l'ondelette de Haar. Cette fois c'est la localisation fréquentielle qui est bonne alors que la localisation temporelle ne l'est pas.

1.7.2.2 Ondelettes de Daubechies

Les ondelettes de Daubechies sont orthogonales et à support compact. Elles permettent de générer des filtres à réponse impulsionnelle finie (RIF) et sont de ce fait très souvent utilisées en imagerie pour la compression.

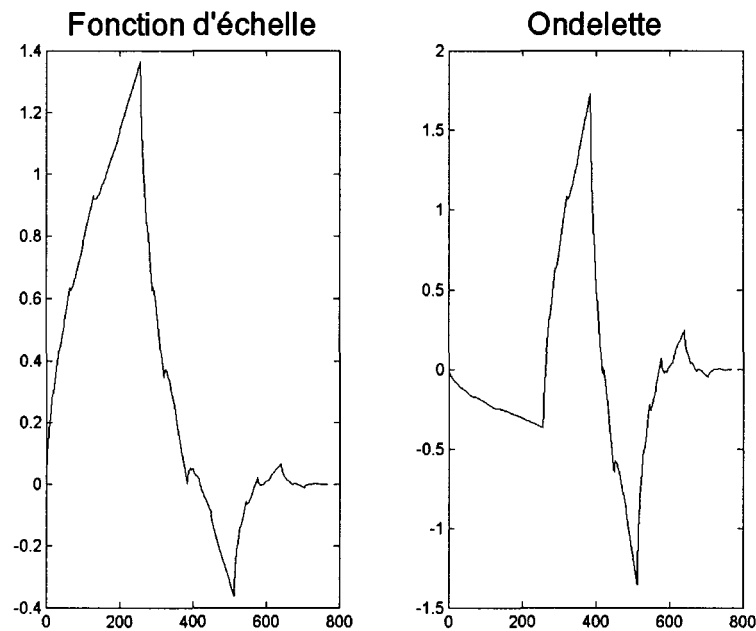


Figure 15 Ondelette de Daubechies D2

1.7.2.3 Ondelettes biorthogonales

Dans le cas des ondelettes biorthogonales, les filtres d'analyse (h_0 et h_1) et de reconstruction (g_0 et g_1) ne sont plus symétriques, mais l'orthogonalité entre filtres croisés est imposée, c'est-à-dire :

$$\begin{aligned} \langle g_0, h_1 \rangle &= 0, \\ \text{et } \langle g_1, h_0 \rangle &= 0. \end{aligned} \tag{1.35}$$

Ceci revient donc à considérer une paire de fonctions d'échelle et ondelettes.

1.7.2.4 Ondelettes de Battle-Lemarié

Ces ondelettes sont créées à partir de B-Splines. Les B-Splines permettent de faire une interpolation cubique d'un signal. Elles conviennent donc bien à une analyse multirésolution.

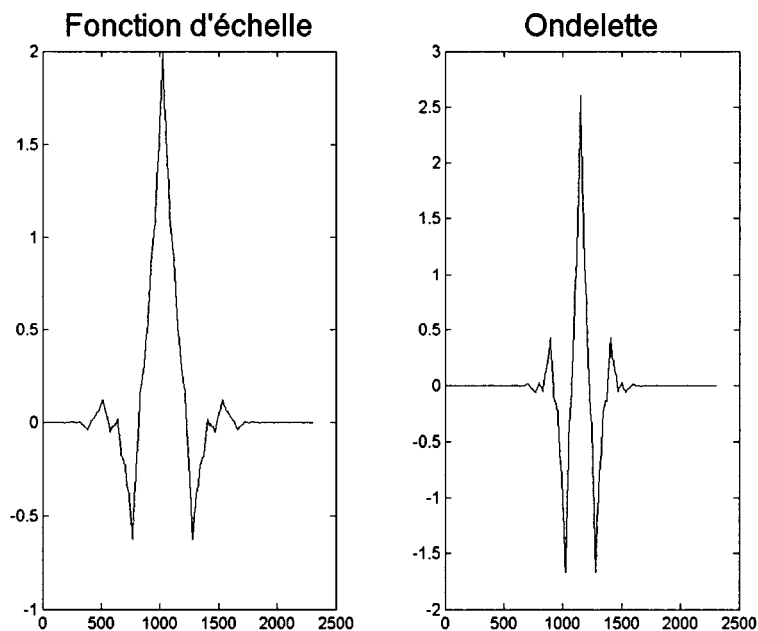


Figure 16 Ondelette biorthogonale 2.4

Battle (Battle, 1987) et Lemarié (Lemarié, 1988) ont utilisé les B-Splines pour générer des ondelettes orthogonales. En revanche, les ondelettes obtenues ne sont pas à support compact.

1.7.3 Ondelettes générées par la méthode du lissage

Daubechies et Sweldens (Daubechies et Sweldens, 1998 ; Sweldens, 1994) proposent une méthode dite de lissage (*lifting* en anglais) qui permet, dans le cas discret, de générer directement des filtres QMF qui possèdent la propriété de pouvoir effectuer une analyse multirésolution. Cette méthode peut également être utilisée pour obtenir une décomposition qui ne donne que des valeurs entières, sans perte d'information.

1.8 Cas multidimensionnel

Dans le cas où on désire effectuer une transformée en ondelettes d'un signal multidimensionnel, typiquement une image ou un volume, des ondelettes séparables sont utilisées¹. Cela revient à effectuer des transformées en ondelettes successivement sur chacune des dimensions.

Pour une image, après une étape de décomposition, quatre sous-images sont donc obtenues. Une est obtenue en passant le filtre passe-bas selon les deux directions. Deux sont obtenues en passant le filtre passe-bas selon une direction et le filtre passe-haut selon l'autre direction. La dernière est obtenue en passant le filtre passe-haut selon les deux directions. La figure 17 illustre la décomposition et les 4 sous-images obtenues.

De façon similaire avec la décomposition d'une image, la représentation de la figure 18 est utilisée pour la décomposition d'un volume. Huit sous-volumes sont alors obtenus. La décomposition peut ensuite être poursuivie sur les voxels de basse fréquence.

Après décomposition, la formule de reconstruction du volume est celle de l'équation 1.36.

$$I_v(\vec{x}) = \sum_{\vec{j} \in V_0} c_{\vec{j}} f_{0,\vec{j}}(\vec{x}) + \sum_{i=1}^7 \sum_{\vec{j} \in V_i} d_{i,\vec{j}} f_{i,\vec{j}}(\vec{x}), \quad (1.36)$$

où $\vec{x} = (x_1, x_2, x_3)^T \in \mathbb{Z}^3$ est la position dans le volume, I_v l'intensité reconstruite, $\{(V_i)_{i \in [0..7]}\}$ l'ensemble des sous-espaces et $\{(f_{i,\vec{j}})_{i \in [0..7], \vec{j} \in V_i}\}$ l'ensemble des fonctions de base obtenues à partir des deux fonctions ϕ et ψ correspondant à l'ondelette utilisée selon l'équation 1.37.

¹ D'autres méthodes peuvent être utilisées, comme les ondelettes en quinconce, mais cela pose un problème de rapidité d'exécution

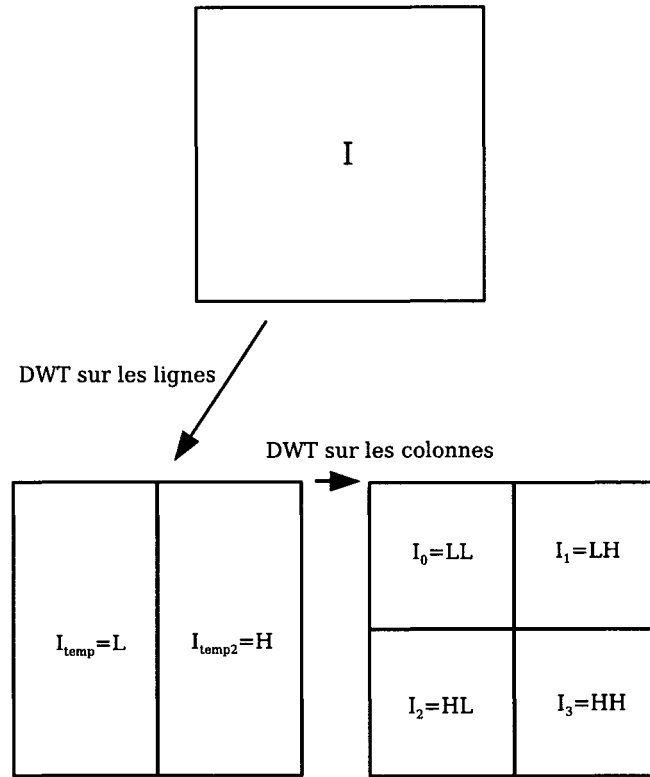


Figure 17 Représentation de la décomposition d'une image

$$\begin{aligned}
 f_{0,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\phi(x_2)\phi(x_3), \\
 f_{1,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\phi(x_2)\psi(x_3), \\
 f_{2,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\psi(x_2)\phi(x_3), \\
 f_{3,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\psi(x_2)\psi(x_3), \\
 f_{4,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\phi(x_2)\phi(x_3), \\
 f_{5,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\phi(x_2)\psi(x_3), \\
 f_{6,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\psi(x_2)\phi(x_3), \\
 f_{7,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\psi(x_2)\psi(x_3), \\
 f_{i,(j_1,j_2,j_3)}(x_1, x_2, x_3) &= f_{i,\vec{0}}(x_1 - j_1, x_2 - j_2, x_3 - j_3).
 \end{aligned} \tag{1.37}$$

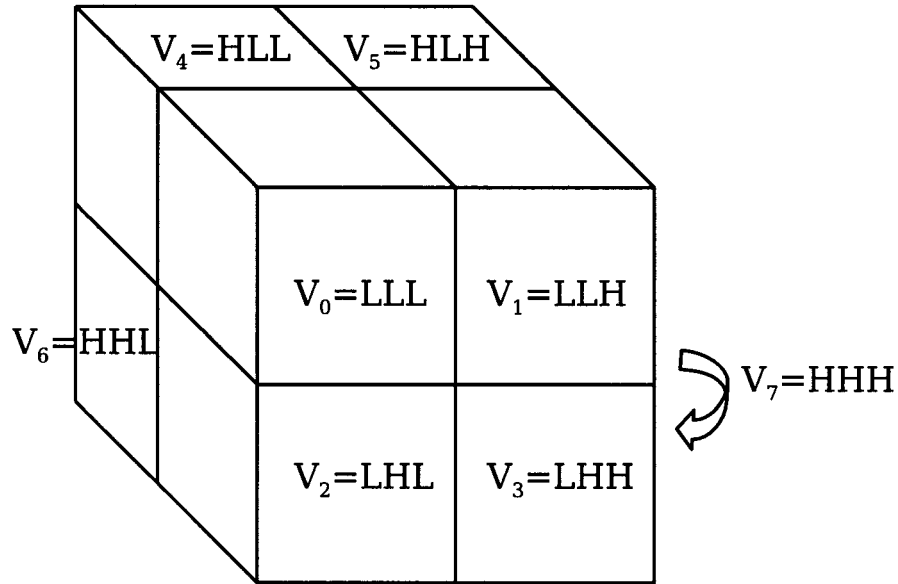


Figure 18 Représentation d'un volume décomposé

Pour une décomposition d'ordre O , c'est-à-dire lorsque la décomposition est répétée sur le volume de basse résolution $O - 1$ fois, la formule de reconstruction devient celle de l'équation 1.38.

$$I_v(\vec{x}) = \sum_{\vec{j} \in V_{O,0}} c_{\vec{j}} f_{O,0,\vec{j}}(\vec{x}) + \sum_{o=1}^O \sum_{i=1}^7 \sum_{\vec{j} \in V_{o,i}} d_{o,i,\vec{j}} f_{o,i,\vec{j}}(\vec{x}). \quad (1.38)$$

où \vec{x} est la position dans le volume, I_v l'intensité reconstruite, O le niveau de décomposition du volume, $\{V_{O,0}, (V_{o,i})_{i \in [1..7]}\}$ l'ensemble des sous-espaces et $\{(f_{O,0,\vec{j}})_{\vec{j} \in V_{O,0}}, (f_{o,i,\vec{j}})_{o \in [1..O], i \in [1..7], \vec{j} \in V_{o,i}}\}$ l'ensemble des fonctions de base obtenues à partir des deux fonctions ϕ et ψ correspondant à l'ondelette utilisée selon les équations 1.39.

$$\begin{aligned}
f_{1,0,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\phi(x_2)\phi(x_3), \\
f_{1,1,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\phi(x_2)\psi(x_3), \\
f_{1,2,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\psi(x_2)\phi(x_3), \\
f_{1,3,\vec{0}}(x_1, x_2, x_3) &= \phi(x_1)\psi(x_2)\psi(x_3), \\
f_{1,4,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\phi(x_2)\phi(x_3), \\
f_{1,5,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\phi(x_2)\psi(x_3), \\
f_{1,6,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\psi(x_2)\phi(x_3), \\
f_{1,7,\vec{0}}(x_1, x_2, x_3) &= \psi(x_1)\psi(x_2)\psi(x_3), \\
f_{O,i,\vec{j}}(x_1, x_2, x_3) &= f_{O,-i,\vec{j}}(2x_1, 2x_2, 2x_3), \\
f_{O,i,(j_1,j_2,j_3)}(x_1, x_2, x_3) &= f_{O,i,\vec{0}}(x_1 - j_1, x_2 - j_2, x_3 - j_3).
\end{aligned} \tag{1.39}$$

1.9 Résumé

En somme nous avons pu voir les raisons pour lesquelles les ondelettes ont été développées. Dans le cadre du rendu de volume, les ondelettes sont essentiellement utilisées pour la compression des données (Bajaj et al., 2001 ; Grosso et al., 1996 ; Kim et Shin, 1999 ; Rodler, 1999) et pour un rendu multirésolution. En effet elles permettent une représentation multirésolution des données utilisant le moins de voxels possible. La partie suivante présentera plus précisément la façon dont les ondelettes sont utilisées. De plus nous verrons que leur propriété de localisation s'avère très utile pour la définition de notre région d'intérêt.

CHAPITRE 2

ÉTAT DE L'ART

Ce chapitre a pour but de faire un tour des méthodes utilisées pour effectuer du rendu de volume sur des données de grande dimension. Les techniques de rendu de volume sont les méthodes qui permettent de visualiser sur un écran (2D) des données qui sont en trois dimensions. Dans le cadre de cette étude, seules les méthodes qui donnent des informations sur la profondeur des différentes couches de nos données sont considérées. Cette information est nécessaire dans le domaine médical lorsque, par exemple, l'utilisateur désire savoir si un corps étranger traverse complètement tel ou tel tissu. Par ailleurs, les techniques de rendu étudiées supposent des données composées de voxels (Drebin et al., 1988) par opposition à des données composées de polygones ou autres formes géométriques.

Deux méthodes de rendu de volume sont généralement utilisées : le rendu de type rayon X, qui affiche une image de type radiographique où toutes les couches sont superposées de façon égales et le rendu avec transparence qui considère certains tissus comme transparents et permet ainsi de voir à travers ces tissus. D'autre part, ces données ayant une résolution importante, il est parfois utile de se limiter au rendu d'une zone particulière du volume appelée région d'intérêt. Nous verrons ainsi les différentes techniques de rendu de volume proprement dites, ainsi que la façon de gérer les régions d'intérêt en encodant d'une façon particulière les données.

2.1 Modèles de rendu

Avant de voir les différentes techniques utilisées pour effectuer du rendu de volume, il convient de préciser les différents modèles utilisés. Ces différents modèles de rendu influent sur ce qui est observable : un ou plusieurs objets, avec ou sans effet de profondeur.

2.1.1 Rendu de volume direct

Le rendu de volume direct ou DVR (pour *Direct Volume Rendering* en anglais) est la technique la plus aboutie pour observer plusieurs objets avec effet de profondeur. Elle permet en effet de gérer la transparence, l'illumination et l'ombrage des différents objets. Pour l'effet de profondeur l'illumination de type Phong-Blinn (Blinn, 1977 ; Phong, 1975) suivante est utilisée :

$$I = I_a + I_d + I_s \quad (2.1)$$

où I_a représente l'intensité de la lumière ambiante. Il s'agit d'une constante. I_d représente la composante diffuse de la lumière. Elle dépend uniquement de la position de la source par rapport à l'orientation de la surface de l'objet.

$$I_d = k_d \langle \vec{N}, \vec{L} \rangle \quad (2.2)$$

où k_d est le coefficient de réflexion diffuse de l'objet considéré, \vec{N} est la normale de la surface et \vec{L} est le vecteur directeur de la direction de la lumière. La normale de la surface est généralement approximée par le gradient du volume. I_s est la composante spéculaire de la lumière. Elle dépend de la position de la source et de l'observateur par rapport à l'orientation de la surface de l'objet.

$$I_s = k_s \cdot \frac{\langle (\vec{V} + \vec{L}), \vec{N} \rangle}{\|\vec{V} + \vec{L}\|} \quad (2.3)$$

où k_s est le facteur spéculaire de l'objet considéré et \vec{V} est le vecteur directeur de la direction d'observation.

Pour observer différents objets, on ajoute un terme d'opacité. Cette opacité est ensuite intégrée afin de bien observer la position relative des objets. Ainsi il est par exemple possible de voir une peau transparente avec les os opaques. Cette intégration s'effectue comme suit :

$$I(\vec{w}, \vec{n}) = \int_{SL}^{SF} I_v(\vec{w} + s\vec{n}) e^{-\int_{SL}^s \xi(\vec{w} + s'\vec{n}) ds'} ds' \quad (2.4)$$

où $\vec{w} = (u, v, 0)^T$ est la position sur le plan d'observation, $\vec{n} = (0, 0, 1)^T$ est la normale du plan d'observation, SL est le point d'entrée dans le volume du rayon lumineux, SF son point de sortie, $I_v(\vec{w} + s\vec{n})$ l'intensité du volume à la position $\vec{w} + s\vec{n}$ le long du rayon lumineux, et $\xi(\vec{w} + s\vec{n})$ la transparence du voxel à la position $\vec{w} + s\vec{n}$ le long du rayon lumineux (figure 19).

En choisissant un pas d'intégration suffisamment petit, Lippert (Lippert, 1998) montre que l'intégration s'effectue selon l'équation suivante :

$$\begin{aligned} I(\vec{w}, \vec{n}) &= \sum_{i=SL}^{SF} \left(I_v(\vec{w} + i\vec{n}) \prod_{j=SL}^{i-1} \xi(\vec{w} + j\vec{n}) \right) \\ &= \sum_{i=SL}^{SF} \left(I_v(\vec{w} + i\vec{n}) \prod_{j=SL}^{i-1} (1 - \alpha(\vec{w} + j\vec{n})) \right), \end{aligned} \quad (2.5)$$

où $\alpha(\vec{w} + j\vec{n}) = 1 - \xi(\vec{w} + j\vec{n})$ est l'opacité du voxel à la position $\vec{w} + s\vec{n}$.

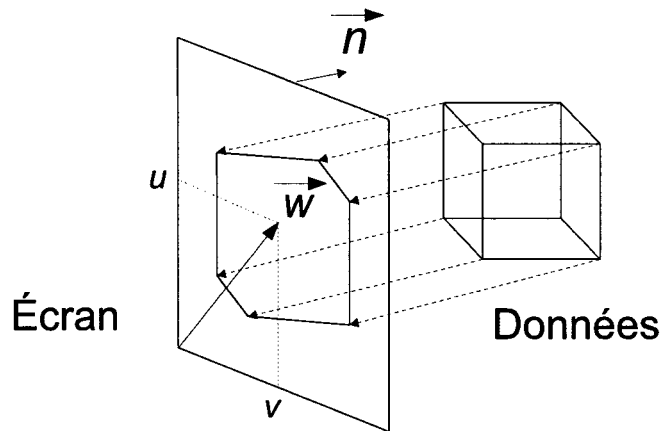


Figure 19 Projection des données

2.1.2 Rendu radiographique

Le rendu radiographique consiste à effectuer une simple intégration de l'intensité du volume selon une direction donnée, sans considérer le facteur d'opacité. L'équation 2.4 est

simplifiée et devient donc :

$$I(\vec{w}, \vec{n}) = \int_{SL}^{SF} I_v(\vec{w} + s\vec{n}) ds \quad (2.6)$$

C'est la méthode la plus simple pour voir tous les objets d'un volume. En contre partie, il est plus difficile de les distinguer. En outre, si le volume n'est pas pré-traité avec des calculs d'illumination, le rendu ne donne pas d'information de profondeur. Seul le fait de pouvoir le tourner donne cette information.

2.1.3 Projection du maximum d'intensité

Dans le cas du MIP (pour *Maximum Intensity Projection* en anglais), seuls les voxels d'intensité maximale sont projetés sur l'écran. On ne peut donc voir que les objets dont l'intensité est maximale et on n'a aucune information de profondeur. Cette méthode est souvent utilisée pour visualiser les vaisseaux sanguins car cela suffit généralement à effectuer un diagnostic. Bien que ne tenant pas compte de l'épaisseur des objets, cette méthode ne diffère des deux méthodes précédentes que par l'équation du calcul de l'intensité, qui est la suivante :

$$I(\vec{w}, \vec{n}) = \max_{s \in [SL, SF]} (I_v(\vec{w} + s\vec{n})) \quad (2.7)$$

La figure 20 montre les mêmes données rendues avec un modèle différent.

2.2 Techniques de rendu de volume

Une fois le modèle de rendu déterminé, il existe plusieurs façons de calculer l'intensité en chaque point du plan d'observation : par projection de voxels sur l'écran, par lancé de rayon, par l'algorithme de *shear-warp factorization*, par application de textures, par projection d'empreintes d'ondelettes ou par une combinaison des précédentes méthodes. Meissner (Meißner et al., 2000) effectue une comparaison plus approfondie en terme de

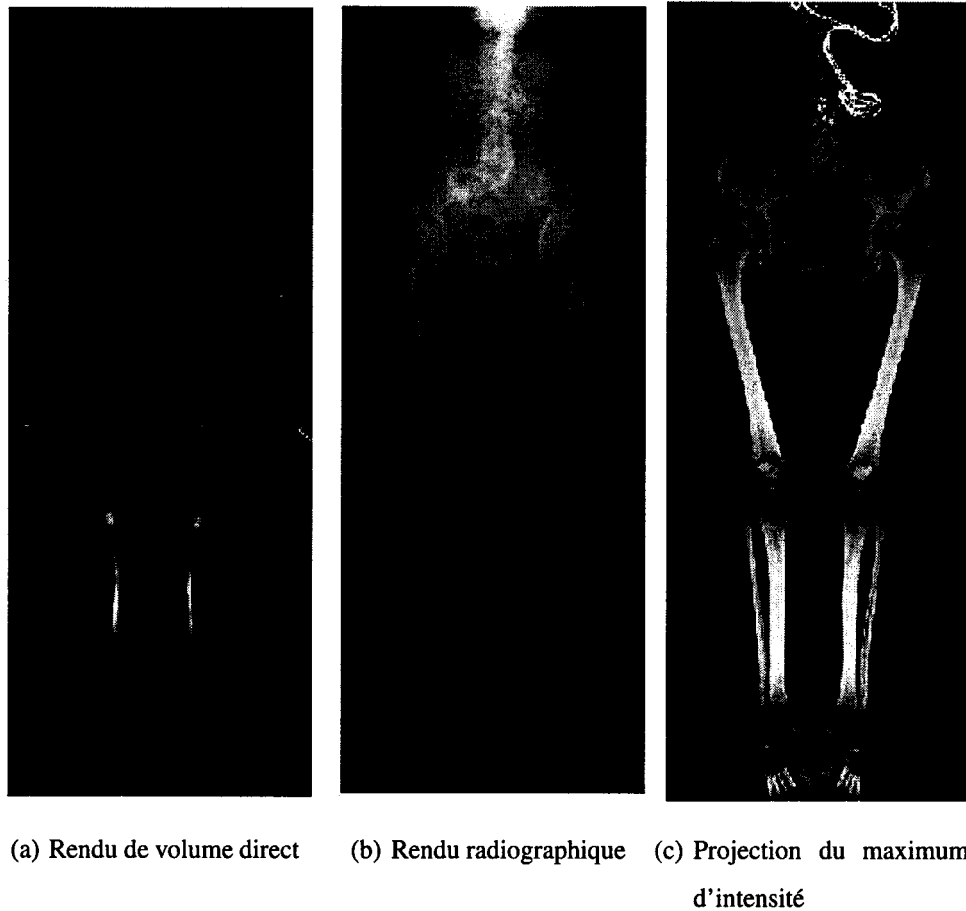


Figure 20 Données composées de 1024 tranches de 512×512 pixels rendues selon le même point de vue avec un rendu différent

performances des méthode de lancé de rayon, projection de voxels, *shear-warp* et des techniques à base de texture.

2.2.1 Lancé de rayon

La première méthode, proposée par Kajiya (Kajiya et Herzen, 1984) et améliorée par Levoy (Levoy, 1988, 1990), consiste à parcourir les données du plan de projection à l'objet. À partir de chaque pixel du plan de projection, un rayon est lancé en direction de l'objet. Le volume est échantillonné à intervalle régulier le long de ce rayon et les voxels du volume sont interpolés pour obtenir la valeur aux points d'échantillonnages. Ceux-ci sont

alors assemblés selon le modèle d'illumination choisi. Le fait de partir de l'image pour parcourir le volume permet de terminer plus tôt l'intégration des échantillons, lorsque cette intégration ne varie plus à cause de l'opacité. Cette technique est mieux connue sous son terme anglais *early ray termination*. En contre partie il est assez difficile de gérer les zones creuses du volume qui ne contribuent pas à l'image finale. Des techniques ont été proposées pour gérer ces espaces creux, comme l'utilisation de boîtes, d'arbres octales ou des approches multirésolutions.

Une autre amélioration possible est l'utilisation de la cohérence spatiale dans l'espace image. En effet, Ihm (Ihm et Lee, 1995) a constaté qu'il est statistiquement fort probable que si deux pixels ont la même couleur, les pixels se situant entre les deux auront également cette couleur. Ceci permet de sous-échantillonner notre espace image et de ne calculer les pixels intermédiaires qu'en cas de non cohérence. Moins de rayons sont ainsi lancés.

Dans la même idée, on peut utiliser la cohérence spatiale dans l'espace objet (Ihm et Lee, 1995). Une décomposition hiérarchique du volume permet alors de réduire le nombre d'échantillons à calculer le long des rayons, les parties homogènes étant traversées d'une traite.

2.2.2 Projection de voxels sur le plan de projection

Plus connue sous son terme anglais, *splatting*, cette seconde méthode très simple consiste à projeter les voxels du volume sur le plan de projection, c'est à dire de parcourir les données depuis le volume jusqu'au plan de projection. Les voxels sont au préalable convolués par des empreintes qui permettent de faire leur interpolation. Une première évaluation de ces empreintes a été proposée par Westover (Westover, 1990). Il a montré que des empreintes gaussiennes permettaient d'éviter des artefacts car les gaussiennes se chevauchent de façon douce.

Des améliorations ont bien entendu été proposées pour améliorer la qualité et la rapidité du rendu. La qualité peut être améliorée en utilisant des empreintes différentes selon la direction de projection. Zwicker (Zwicker et al., 2002) propose ainsi d'utiliser des empreintes gaussiennes elliptiques. De même, Swan (Swan et al., 1997) propose une méthode pour limiter le crénelage. Une autre façon d'améliorer la qualité du rendu et d'utiliser cette technique conjointement avec une autre. Cai (Cai et Sakas, 2000) et Mora (Mora et al., 2002) proposent ainsi de l'utiliser avec la méthode du lancé de rayon, tandis que Mueller (Mueller et Yagel, 1996) l'utilise conjointement avec la technique *Shear-Warp*.

En ce qui concerne la rapidité, on peut utiliser les capacités des cartes graphiques (Chen et al., 2004) ainsi que des structures adéquates pour traiter rapidement le volume (en occultant entre autres les espaces vides) (Ihm et Lee, 1995 ; Kiltath et Möller, 2001 ; Orchard et Moller, 2001). De même des méthodes de rendu progressif permettent également d'avoir un rendu plus rapide (Laur et Hanrahan, 1991). Dans certains cas, l'utilisation d'une post-convolution peut également accélérer le rendu (Neophytou et Mueller, 2003).

2.2.3 Méthode *Shear-Warp*

Il s'agit d'une méthode astucieuse proposée par Lacroute (Lacroute et Levoy, 1994) pour profiter des avantages des deux méthodes précédentes. Elle consiste à décaler au préalable les images puis de faire la composition directement sur ces images. Une opération 2D est finalement appliquée pour rétablir un système de coordonnées cohérent (figure 21).

Le fait de faire le rendu sur les images décalées permet de n'effectuer que des interpolations bilinéaires dans l'image comme dans le cas de la projection de voxels plutôt que trilineaires dans le volume comme dans le tracé de rayon. D'autre part elle permet un encodage efficace des données de type *run-length*. Il est également facile d'implémenter la terminaison rapide de l'intégration de l'opacité (*early ray termination*). Cette méthode est réputée pour être la plus rapide à s'exécuter de façon logicielle. Pour améliorer la qualité du rendu Sweeney (Sweeney et Mueller, 2002) propose d'utiliser une classification et une

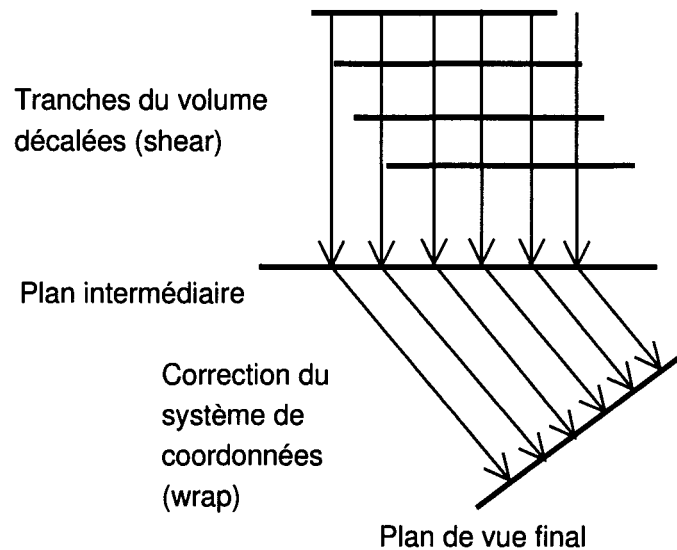


Figure 21 Résumé de la méthode *Shear-Warp*

illumination après l'interpolation des données alors qu'à la base celle-ci se fait avant et peut entraîner des artefacts. Il propose également d'ajuster le pas d'intégration selon la résolution désirée et d'utiliser des tranches intermédiaires dans certains cas pour éliminer des effets de crénelage. De même Schulze (Schulze et al., 2003) propose une méthode pour utiliser cette technique avec une pré-intégration du volume en minimisant la perte de qualité.

2.2.4 Utilisation du matériel graphique

Les techniques de rendu étant assez lentes et le matériel graphique se faisant plus accessible, des méthodes spécifiques (Cabral et al., 1994 ; Gelder et Kim, 1996 ; Guthe et Strasser, 2004 ; Guthe et al., 2002 ; Kruger et Westermann, 2003 ; Meißner et al., 2001 ; Rezk-Salama et al., 2000 ; Westermann et Sevenich, 2001 ; Westermann et Ertl, 1998) utilisant les capacités de ce type de matériel ont été développées.

2.2.4.1 Textures 2D

Cette méthode (Rezk-Salama et al., 2000) ainsi que celle qui suit exploite les possibilités matérielles des cartes graphiques relativement récentes. Au lieu d'intégrer voxels par voxels le long des rayons, on intègre par plan. Plus exactement, le volume est décomposé en tranches. Ces tranches sont chargées en tant que textures (avec information d'opacité) et sont ensuite accumulées de la plus lointaine à la plus proche. Pour avoir la possibilité d'effectuer l'intégration dans toutes les directions sans avoir trop d'artefacts, les plans doivent être générés selon les 3 axes principaux (figure 22). La pile de plans dont la direction est la plus proche de celle d'observation est ensuite choisie pour effectuer l'intégration.

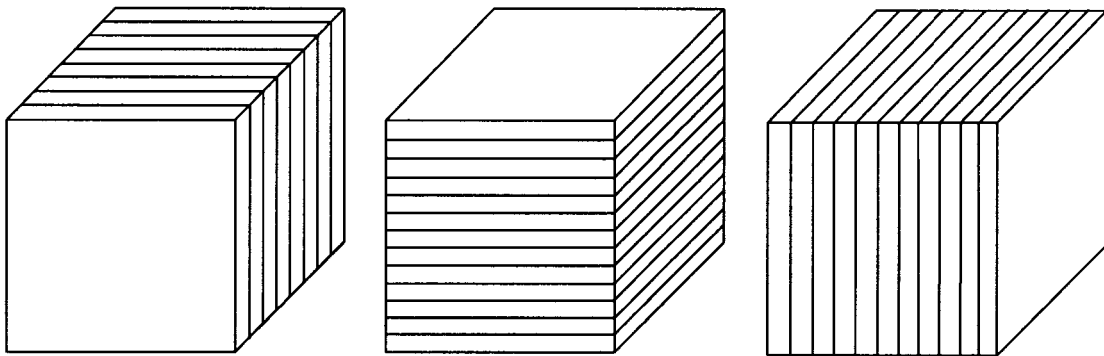


Figure 22 Les trois piles de textures nécessaires

L'avantage réside dans la rapidité d'exécution du rendu. En effet le matériel graphique est spécialisé pour faire ces calculs très rapidement. En contre partie les voxels ne sont interpolés que bilinéairement (sur chaque texture) et l'occupation mémoire nécessite trois fois celle nécessaire au volume original.

2.2.4.2 Textures 3D

Cette méthode, proposée par Cabral (Cabral et al., 1994), est devenue possible par les cartes graphiques supportant les textures tridimensionnelles (NVIDIA geforce 3, ATI radeon 8500 et plus). Elle permet de s'affranchir des deux problèmes majeurs du rendu à

l'aide de textures 2D. En effet l'interpolation des voxels est désormais trilinéaire et une seule copie du volume original est nécessaire. Les données sont chargées dans la mémoire de la carte graphique sous forme de texture 3D. À partir de cette texture, des polygones perpendiculaires à la direction d'observation sont générés (figure 23). Ces polygones sont ensuite ajoutés comme s'il s'agissait de textures 2D. Bien que légèrement moins rapide que la méthode précédente, le rendu obtenu est de meilleure qualité et demeure en temps réel.

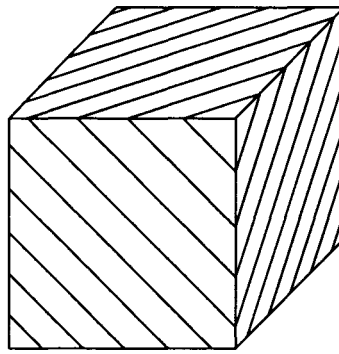


Figure 23 Génération des polygones dans l'axe d'observation

Les limites se situent dans la taille du volume que l'on peut stocker dans la mémoire de la carte graphique. Pour cette raison, il est nécessaire de compresser les données originales. Le volume peut également être découpé en blocs. Ces blocs sont alors rendus indépendamment puis composés de façon logicielle. Cela nécessite par contre plus d'accès mémoire pour charger successivement les différents blocs dans la mémoire de la carte graphique. Le coût de ces accès entraîne une chute drastique des performances de telle sorte que la méthode *shear-warp* peut s'avérer plus efficace (Meißner et al., 2000).

2.2.5 Techniques à base d'ondelettes

Les techniques à base d'ondelettes permettent de travailler directement sur le volume compressé. La compression se fait suite à la transformée en ondelettes du volume. On peut alors utiliser une compression en effectuant un seuil sur les coefficients obtenus (ceux

étant proches de zéro contribuant peu au volume final). D'autre part des méthodes efficaces de codage des coefficients d'ondelettes peuvent être utilisées comme l'EZW (pour *Embedded Zerotree of Wavelets* en anglais) ou le SPIHT (pour *Set Partitional In Hierarchical Tree* en anglais). De nombreuses méthodes (Guthe et Strasser, 2004 ; Guthe et al., 2002 ; Pinnamaneni, 2003 ; Saladi et al., 2003 ; Sanchez et Carvajal, 1998) reconstruisent le volume à une certaine résolution avant d'effectuer un des rendus cités précédemment.

Gross et Lippert (Gross et al., 1995, 1994 ; Lippert, 1998 ; Lippert et Gross, 1995) sont à l'origine d'une méthode originale appelée *Wavelet Splatting*. Pour effectuer le rendu, au lieu de projeter les voxels sur l'écran, on projette des empreintes pondérées d'ondelettes. Ces empreintes correspondent aux projections des 8 types d'ondelettes selon la direction d'observation. Cette méthode a pour limite de ne pas pouvoir intégrer la transparence dans le rendu et nécessite de précalculer l'éclairement sur le volume si on ne veut pas d'un rendu radiographique. Toutefois elle est relativement rapide car la représentation originale du volume permet de le compresser et cela ne nécessite pas ensuite de reconstruire le volume avant d'effectuer le rendu. Cette méthode a ensuite été améliorée par Horbelt (Horbelt et al., 1999) qui propose une autre méthode pour générer les empreintes, et par Westenberg (Westenberg et Roerdink, 2000) qui propose de projeter directement les coefficients d'ondelettes avant de les convoluer par les empreintes.

2.2.6 Techniques hybrides

Il existe également des techniques (Hadwiger et al., 2003), permettant de rendre les tissus avec des techniques différentes, par exemple, les vaisseaux sanguins et la peau en MIP, les organes et les os avec du lancé de rayon. Les images obtenues sont de très bonnes qualité mais l'inconvénient majeur réside dans le fait qu'il faut que le volume soit segmenté au préalable. Cette dernière tâche est très difficile et longue à effectuer et nécessite une intervention manuelle.

2.3 Représentation du volume avec volume d'intérêt

Étant donné la croissance de la définition des méthodes d'acquisitions des données (scanners, imagerie par résonance magnétique), les volumes sont composés de plus en plus de pixels et occupent une place mémoire de plus en plus importante. Il n'est en effet pas rare d'obtenir des données de plus de $512 \times 512 \times 512$ voxels, codées sur plus de 16 bits. Ceci prend alors déjà 256 Mo. Si on considère que l'on a parfois besoin de plusieurs copies du volume (pour ajouter des informations telles l'opacité, le gradient), la gestion de la mémoire devient un problème. Une solution est donc de compresser les données. Outre les méthodes traditionnelles de compression (run-length, Huffman, codage arithmétique), une des méthodes les plus utilisées actuellement utilise la transformée en ondelettes. Cette transformée en elle-même ne compresse pas le volume puisque bijective, mais permet d'appliquer différentes méthodes de compression de manière efficace (Grosso et al., 1996).

D'autre part, dans la plupart des cas, il n'est pas nécessaire d'effectuer le rendu du volume en entier. En effet, seule une petite partie du volume est intéressante pour effectuer un diagnostic ou pour la surveillance de l'opération. Cette partie du volume est appelée région d'intérêt et peut couvrir moins de 10% du volume initial. Il est donc très intéressant de profiter de cette observation : les calculs de rendu étant longs, on peut essayer de les limiter à cette zone. Toutefois la zone qui entoure notre volume d'intérêt, qui est appelée zone contextuelle ou plus simplement contexte est tout de même nécessaire pour bien situer notre volume d'intérêt. Un bon compromis est donc d'utiliser une approche multirésolution pour laquelle la résolution dépend de la zone dans laquelle on se situe : une haute résolution pour la zone d'intérêt et une faible résolution pour la région contextuelle. Pour cette approche multirésolution, les ondelettes conviennent parfaitement puisqu'elles permettent en outre d'avoir une représentation compacte (contrairement à d'autres méthodes comme une représentation pyramidale (LaMar et al., 2000)).

2.3.1 Codage par blocs

Souvent la région d'intérêt est connue à l'avance et on peut donc enregistrer directement le volume avec cette propriété. L'inconvénient est que l'information des détails pour la zone contextuelle est perdue. Ceux-ci peuvent être intéressants si la zone d'intérêt est déplacée ou agrandie.

Le volume peut être codé par blocs (Guthe et Strasser, 2004 ; Guthe et al., 2002 ; Krishnan et al., 2004). Chaque bloc est alors compressé de façon différente selon qu'il contribue à la région d'intérêt ou non. Ainsi les blocs contribuant uniquement à la zone contextuelle peuvent être fortement compressés, quitte à perdre en définition, tandis que les blocs contribuant à la région d'intérêt doivent subir le moins de perte possible.

On peut constater deux inconvénients majeurs à cette méthode d'encodage par blocs. D'une part, on n'a pas de liberté au niveau de la zone d'intérêt. Ceci ne pose pas de problème uniquement si cette zone a été préalablement identifiée, ce qui n'est pas toujours le cas. D'autre part utiliser des blocs peut induire des problèmes en bordure de blocs. En effet si on utilise une compression par ondelette, la reconstruction des bords n'est pas parfaite. Il faut soit utiliser les données autour de ce bord, soit utiliser des fonctions d'ondelettes supplémentaires pour gérer les bords. La deuxième solution ne permet pas une décompression rapide sous forme de filtres. Pour remédier à ce problème, il faut donc soit que les blocs se chevauchent, auquel cas on a une redondance d'information, soit faire un rendu en chargeant les blocs voisins, ce qui utilise inutilement de la mémoire.

Or la propriété de localisation spatiale des ondelettes permet de passer outre cette décomposition par blocs. En effet chaque coefficient d'ondelettes n'agit que sur une petite portion du volume, portion dont la taille dépend du support de l'ondelette choisie. Ainsi une décomposition en ondelettes sur l'ensemble du volume permet de faire l'équivalent d'une décomposition par blocs. Comme la décomposition se fait sur l'ensemble du volume on n'a plus de soucis d'effets de bords car les bords du volume sont un espace vide.

2.3.2 Codage par arbres

Cette représentation est utilisée car on a constaté qu'il y a une forte corrélation spatiale entre les coefficients d'ondelette des données de niveau inférieur (basse résolution) et ceux des niveaux supérieurs (détails). Ainsi lorsque les coefficients de basse résolution sont faibles, il est fort probable que les détails correspondant à la même zone le soient aussi. On peut alors éliminer toute une partie des coefficients correspondants à la même zone.

La méthode EZW (*Embedded Zerotree of Wavelets*) développée par Shapiro (Shapiro, 1993) permet d'encoder les coefficients. Cet encodage a la particularité de pouvoir être aisément décodé de façon partielle selon la précision (le nombre de coefficients) que l'on désire pour reconstruire le volume. En effet les coefficients les plus forts, donc ceux qui ont le plus d'influence sur le volume sont stockés au début de la structure. Il est dès lors possible de s'arrêter à n'importe quel moment de lire les données, une représentation plus ou moins précise du volume sera obtenue.

La méthode SPIHT (*Set Partitional In Hierarchical Tree*) (Said et Pearlman, 1996) s'inspire de la précédente. Elle modifie la façon dont les coefficients sont ordonnancés. Les coefficients les plus importants sont stockés au début de la structure. Ainsi ce ne sont pas forcément toujours les coefficients de plus basse définition qui sont stockés au début. On peut par exemple donner de l'importance aux coefficients permettant de reconstruire une région du volume.

Le désavantage de ces deux techniques est que la zone d'intérêt doit être connue à l'avance pour que l'encodage fonctionne. Toutefois nous verrons qu'en utilisant quelques informations supplémentaires, un codage par arbre peut s'avérer très utile pour notre objectif.

2.4 Travaux proches

Lippert (Lippert, 1998) propose une méthode de rendu qui se fait directement à partir du volume transformé en ondelettes. Ceci présente l'avantage de ne pas avoir à gérer des problèmes de bord, de ne pas avoir à reconstruire le volume et de pouvoir effectuer le rendu de manière progressive sans avoir à défaire ce qui a été fait auparavant. Leur méthode ne profite cependant pas complètement de la notion de région d'intérêt. Celle-ci est uniquement utilisée pour compresser les données et est donc fixée lors de la transformée en ondelettes du volume original. En outre, la festion des tissus transparents en limite. L'opacité des tissus ne peut pas être intégrée lors du rendu proprement dit. De même, les données originales doivent être isotropes. Pour combler ces deux problèmes des prétraitements sont nécessaires.

Krishnan (Krishnan et al., 2004) propose d'utiliser une représentation des données en adéquation avec la norme JPEG2000. Les données sont décomposées sous forme de blocs qui sont décomposés en ondelettes. Leur application est une application client-serveur où les données sont stockées sur le serveur et transmises au client via un réseau. La région d'intérêt est sélectionnée par le client et le serveur transmet les blocs nécessaires au rendu. Les blocs contribuant à la région d'intérêt sont intégralement transmis tandis que les blocs contribuant au contexte sont transmis à faible résolution. Le problème intervient au moment du rendu. En effet il est alors nécessaire de reconstruire chaque bloc avant d'effectuer le rendu. Cette reconstruction prend du temps, d'autant plus qu'il faut gérer les bords de ces blocs (section 2.3.1).

D'autre part Lamar (LaMar et al., 2000) propose une méthode avec région d'intérêt non fixe. Cette méthode est basée sur un rendu hardware à base de textures et permet un rendu multirésolution. La multirésolution est obtenue en utilisant une structure pyramidale, les données de basse résolution étant obtenues en faisant la moyenne des données de plus haute résolution. Ceci induit un coût supplémentaire en mémoire puisque l'information

des différentes résolutions doit être conservée. Une représentation à base d'ondelettes devrait être plus adéquate puisqu'elle ne nécessite pas de stocker plus d'information que le volume original. D'autre part cette méthode est limitée lorsque de trop gros volumes sont à traiter puisque les données ne peuvent alors pas être toutes chargées dans la mémoire de l'unité graphique. Il y a alors beaucoup de temps de chargement et déchargement des données. En outre, lorsque la résolution est modifiée, le travail accompli n'est pas réutilisé, le rendu doit être effectué depuis le début.

2.5 Résumé

Les méthodes de visualisation les plus rapides actuellement utilisent les capacités du matériel graphique. Cependant, lorsque les données sont trop importantes en occupation mémoire, il faut découper les données en sous-volumes et trop de temps est passé à charger ou décharger ces données dans la mémoire du matériel graphique. Les méthodes logicielles peuvent alors être plus efficaces. Une façon de gérer cet accroissement en occupation mémoire des données est d'utiliser le rendu multirésolution. Une basse résolution est utilisée lors de la navigation et une haute résolution est utilisée lorsque le point de vue de l'utilisateur est fixé. L'utilisation d'une région d'intérêt permet en plus d'obtenir une navigation interactive tout en gardant une pleine résolution sur une région particulièrement intéressante et d'ainsi définir plus facilement le point de vue le plus intéressant pour l'analyse des données. La représentation multirésolution des données peut être obtenue par structure pyramidale ou par décomposition en ondelettes. Cette dernière méthode à l'avantage d'avoir une représentation plus compacte que la structure pyramidale. Le rendu par projection d'empreintes d'ondelettes permet d'éviter la reconstruction des données mais sacrifie légèrement la qualité du rendu en limitant la gestion des tissus transparents.

CHAPITRE 3

MÉTHODE PROPOSÉE

Ce chapitre a pour but de présenter les idées proposées pour réaliser ce projet, ainsi que les choix techniques adoptés. Pour accélérer le rendu, on pense sacrifier de l'information sur une grande partie du volume, tout en conservant le maximum d'information pour une petite zone d'intérêt. Les propriétés de la transformée en ondelettes que nous avons vues dans le chapitre précédent, à savoir sa représentation compacte et locale d'un volume, nous amènent à utiliser une technique de rendu exploitant cette décomposition. En effet la représentation compacte permet de conserver, voire diminuer, l'occupation mémoire des données et la représentation locale permet d'extraire rapidement une région d'intérêt. Pour éviter d'avoir à reconstruire localement les données, il est tout naturel de se pencher vers la technique proposée par Lippert et Gross et améliorée par Westenberg, à savoir la projection d'empreintes d'ondelettes.

3.1 Calculs préliminaires pour le rendu

La technique de projection des empreintes d'ondelettes a pour limite de ne pas pouvoir intégrer l'opacité des différents tissus lors du rendu proprement dit. Ainsi l'opacité n'est pas intégrée selon la direction d'observation. Cependant, afin d'avoir une meilleure impression de volume, l'illumination est précalculée. Ainsi une préintégration de l'opacité du volume selon une direction fixe est utilisée.

3.1.1 Illumination

Le modèle d'illumination permet d'avoir une meilleure impression de volume. Nous avons choisi une représentation simplifiée du modèle de Phong-Blinn (Blinn, 1977 ; Phong, 1975). En effet la composante spéculaire des données est supprimée car celle-ci dépend

de la position relative de la direction de l'observation et de la direction de la source lumineuse. Or la direction de l'observation ne sera pas fixe dans notre cas puisque les données doivent pouvoir être manipulées. Ceci peut alors engendrer quelques artefacts, notamment des zones lumineuses au mauvais endroit, et ainsi induire en erreur l'utilisateur sur l'orientation des objets du volume. Le modèle utilisé est donc celui présenté à l'équation 3.1.

$$I_v = I_a + I_d = I_a + k_d \langle \vec{L}, \vec{N} \rangle \quad (3.1)$$

où I_v est l'intensité calculée, I_a est la composante ambiante de la lumière et I_d la composante diffuse. La constante k_d est le coefficient de diffusion, L est la direction de la source lumineuse et N est la normale de l'objet considéré, déterminée à partir du gradient lui-même calculé avec un filtre de Sobel 3D. L'approximation du gradient par le filtre de Sobel 3D semble en effet donner des résultats légèrement meilleurs que l'approximation par une simple différence centrée, notamment du point de vue du crénelage (figure 24).

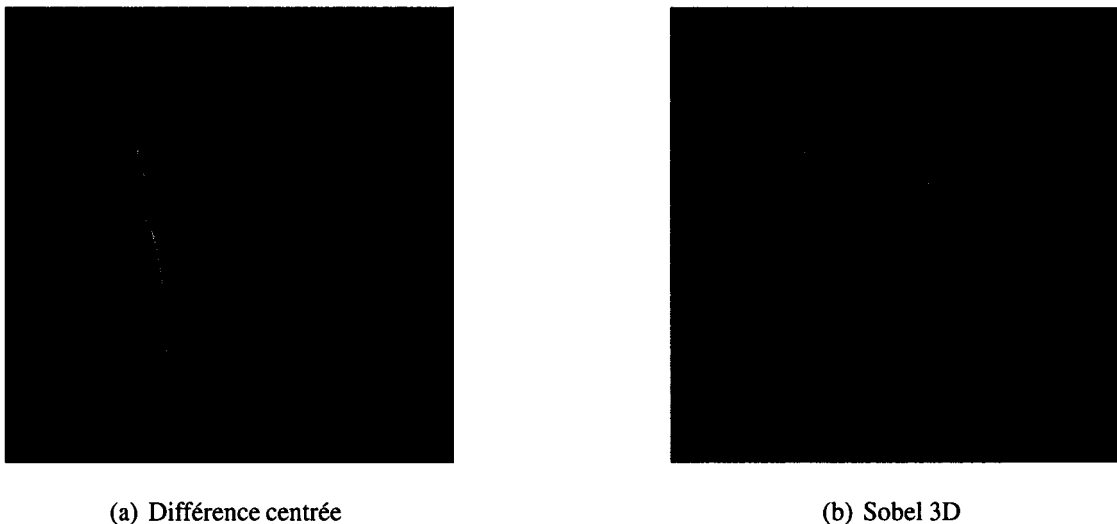


Figure 24 Approximation du gradient par deux méthodes

3.1.2 Classification

Il s'agit de définir l'opacité, α , et le coefficient de diffusion, k_d , des voxels selon leur appartenance à un tissu ou un autre. La segmentation des tissus étant un problème à part entière, la fonction permettant d'associer une valeur du volume à une opacité est simplement linéaire par morceaux. Ce terme permet de percevoir les tissus comme plus ou moins transparents et également d'obtenir des ombres. Pour simplifier, le coefficient de diffusion est choisi constant pour tous les tissus.

3.1.3 Ombrage

L'ombrage est précalculé en associant une opacité aux données et en effectuant leur intégration. Pour cela, une méthode de lancé de rayon est utilisée, comme Levoy (Levoy, 1990) le propose. En choisissant la direction d'intégration comme un des trois axes principaux du volume et en prenant un pas d'intégration de 1, il n'est pas nécessaire d'effectuer une quelconque interpolation des données. L'intégration s'effectue alors comme décrit dans l'équation 3.2 pour chaque voxel de l'image le long d'un axe.

$$\begin{aligned} I_{v,nouveau}(\vec{x}) &= I_v(\vec{x})(1 - \alpha_{temp})\alpha(\vec{x}), \\ \alpha_{temp} &= \alpha_{temp} + (1 - \alpha_{temp})\alpha(\vec{x}). \end{aligned} \tag{3.2}$$

où $\vec{x} = (i, j, k)$ est la position du voxel dans les données, $I_v(\vec{x})$ est l'intensité du voxel une fois le modèle d'illumination appliqué (équation 3.1), $\alpha(\vec{x})$ est l'opacité associée à la valeur originale de ce même voxel. L'opacité varie entre 0, pour un voxel transparent et 1, pour un voxel opaque. La variable temporaire permettant d'intégrer l'opacité, α_{temp} , est initialisée à 0 en début de chaque rayon lancé.

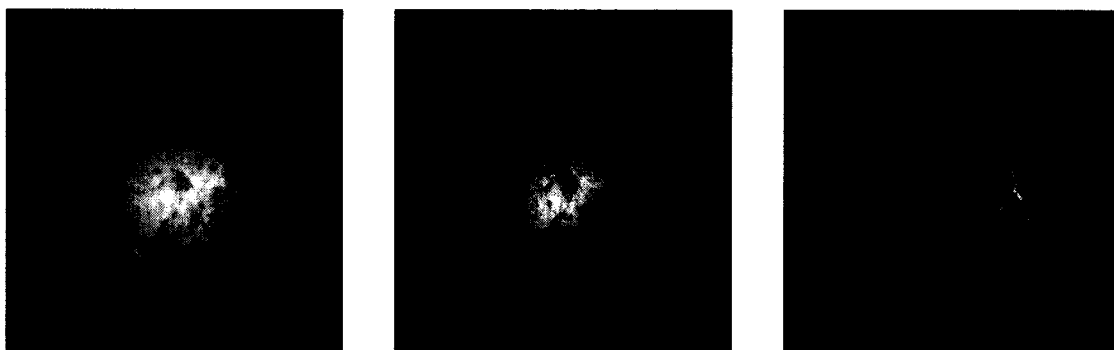
Au final l'algorithme de préintégration peut s'écrire comme dans l'algorithme 1.

Algorithme 1 : Illumination et préintégration de l'opacité selon la profondeur

```

pour  $i \leftarrow 1$  à largeur faire
  pour  $j \leftarrow 1$  à hauteur faire
     $\alpha_{temp} \leftarrow 0$ ;
    pour  $k \leftarrow 1$  à profondeur faire
       $\vec{x} \leftarrow (i, j, k)$ ;
       $\vec{N} \leftarrow \text{Gradient}(\vec{x})$ ;
       $I_v(\vec{x}) = I_a + I_d(\vec{x})$  (équation 3.1);
       $I_v(\vec{x}) \leftarrow I_v(\vec{x})(1 - \alpha_{temp})\alpha(\vec{x})$ ;
       $\alpha_{temp} \leftarrow \alpha_{temp} + (1 - \alpha_{temp})\alpha(\vec{x})$ ;
  
```

La figure 25 montre en quoi ces précalculs sont importants pour une meilleure visualisation des données.



(a) Rendu X-Ray

(b) Ajout de l'illumination

(c) Ajout de l'ombrage

Figure 25 Amélioration de la qualité du rendu avec des calculs d'illumination et d'ombrage

3.2 Projection des empreintes d'ondelettes

La méthode de projection des empreintes est proposée par Gross et Lippert (Gross et al., 1995, 1994 ; Lippert, 1998 ; Lippert et Gross, 1995) et vient du fait que lorsque l'opacité est considérée comme constante à l'intérieur du volume, ou lorsque cette dernière a été

préalablement calculée, il est possible d'inverser la projection du volume et sa reconstruction.

Rappelons la formule de reconstruction d'un volume (équation 1.38).

$$I_v(\vec{x}) = \sum_{\vec{j} \in V_{O,0}} c_{\vec{j}} f_{O,0,\vec{j}}(\vec{x}) + \sum_{o=1}^O \sum_{i=1}^7 \sum_{\vec{j} \in V_{o,i}} d_{o,i,\vec{j}} f_{o,i,\vec{j}}(\vec{x}). \quad (3.3)$$

où \vec{x} est la position dans le volume, I_v l'intensité calculée, O le niveau de décomposition du volume, $\{V_{O,0}, (V_{o,i})_{i \in [1..7]}\}$ l'ensemble des sous-espaces et $\{(f_{O,0,\vec{j}})_{\vec{j} \in V_{O,0}}, (f_{o,i,\vec{j}})_{o \in [1..O], i \in [1..7], \vec{j} \in V_{o,i}}\}$ l'ensemble des fonctions de base.

Rappelons maintenant l'équation permettant de projeter un volume sur l'écran (équation 2.6), en supposant l'opacité des tissus nulle ou préintégré.

$$I(\vec{w}, \vec{n}) = \int_{s=S_{in}}^{S_{out}} I_v(\vec{w} + s\vec{n}) ds. \quad (3.4)$$

Il s'en suit l'équation 3.5

$$I(\vec{w}, \vec{n}) = \int_{s=S_{in}}^{S_{out}} \sum_{\vec{j} \in V_{O,0}} c_{\vec{j}} f_{O,0,\vec{j}}(\vec{w} + s\vec{n}) + \sum_{o=1}^O \sum_{i=1}^7 \sum_{\vec{j} \in V_{o,i}} d_{o,i,\vec{j}} f_{o,i,\vec{j}}(\vec{w} + s\vec{n}) ds. \quad (3.5)$$

L'intégration et la reconstruction du volume peuvent alors être inversées.

$$I(\vec{w}, \vec{n}) = \sum_{\vec{j} \in V_{O,0}} c_{\vec{j}} \int_{s=S_{in}}^{S_{out}} f_{O,0,\vec{j}}(\vec{w} + s\vec{n}) ds + \sum_{o=1}^O \sum_{i=1}^7 \sum_{\vec{j} \in V_{o,i}} d_{o,i,\vec{j}} \int_{s=S_{in}}^{S_{out}} f_{o,i,\vec{j}}(\vec{w} + s\vec{n}) ds. \quad (3.6)$$

Ceci revient donc à une somme pondérée d'intégrales ne dépendant que de l'ondelette choisie. Ces intégrales sont appelées les empreintes d'ondelettes. Ainsi le rendu de volume

consiste en une accumulation de ces empreintes sur l'écran. Du fait de la relation inter-échelle des fonctions d'ondelettes, toutes ces empreintes peuvent être générées à partir de 8 empreintes originales, en les déplaçant et les sous-échantillonnant.

3.2.1 Avantages de la technique

Cette technique présente deux avantages majeurs. Tout d'abord elle permet d'effectuer le rendu directement à partir de la décomposition en ondelettes du volume. Ainsi, contrairement aux autres méthodes couramment utilisées (lancé de rayon, projection de voxels en utilisant ou non les capacités du matériel graphique), il n'est pas nécessaire de reconstruire, ne serait-ce que localement, le volume à traiter.

D'autre part, cette méthode présente l'avantage de pouvoir facilement améliorer la définition du rendu en ajoutant les empreintes correspondant aux détails. Il n'est ainsi pas nécessaire de tenir compte à nouveau des données de basse résolution, ce qui est le cas des autres méthodes.

3.2.2 Choix de l'ondelette

Cette méthode peut bien entendu être appliquée à de nombreux types d'ondelettes. Pour que la méthode de rendu soit le plus rapide possible, il faut que la projection des empreintes soit relativement rapide et donc que ces empreintes comportent peu de pixels. En effet plus les empreintes comporteront de pixels, puis cela prendra du temps pour projeter ces empreintes.

D'autre part, comme la section suivante le présentera, il est très utile de connaître la formule analytique de l'ondelette pour pouvoir générer les empreintes. Cela élimine les ondelettes de Daubechies d'ordre supérieur ou égal à 2 dont la formule analytique n'est pas connue. Les ondelettes générées à partir de B-splines sont par contre adéquates et ont

montré leur efficacité (Lippert et Gross, 1995 ; Lippert, 1998 ; Westenberg et Roerdink, 2000).

Toutefois dans le cadre du rendu avec région d'intérêt il y a une contrainte supplémentaire et non des moindres : les coefficients contribuant à la région d'intérêt doivent être peu nombreux. Les empreintes doivent donc être aussi petites que possible. En effet si des empreintes de large taille sont utilisées, celles-ci se chevaucheront et plus d'empreintes de détail seront nécessaires au rendu de la région d'intérêt. Pour cette raison l'ondelette dont le support est le plus compact possible, soit l'ondelette de Haar, a été choisie.

Ce choix est renforcé par le fait que l'ondelette de Haar est celle qui permet de reconstruire un volume le plus fidèlement possible, d'après Gargantini (Gargantini et al., 2000). Comme le rendu est effectué en pleine résolution sur la région d'intérêt, l'utilisation d'une ondelette d'ordre supérieure n'améliore la qualité de reconstruction que sur la région contextuelle. Or cette région contextuelle sert à localiser la région d'intérêt et la qualité de reconstruction obtenue avec l'ondelette de Haar s'avère suffisante.

3.2.3 Projection utilisée

La méthode de projection d'empreintes d'ondelettes n'autorise que la projection parallèle. Ceci peut poser problème puisqu'une telle projection suppose que l'observateur est placé à une distance infinie des données. Dans notre cas, la taille de l'écran de visualisation est d'environ 15cm. Or, l'observateur est placé à environ 50cm de l'écran. L'écart maximal entre les angles de visualisation est alors d'environ 30° (figure 26). Dans cette situation, la projection parallèle est suffisamment appropriée.

3.2.4 Génération des empreintes d'ondelettes

Un point important du rendu de volume basé sur la projection des ondelettes est la génération de ces projections, appelées empreintes.

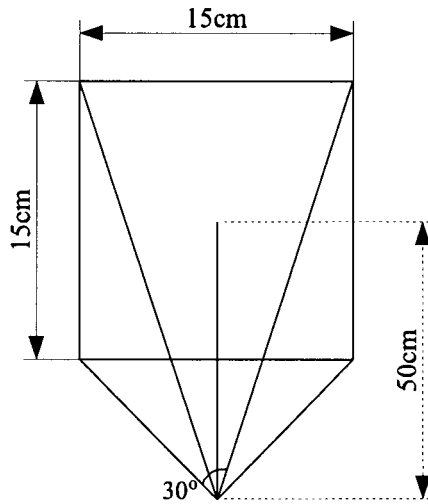


Figure 26 Angle de vue lors de la visualisation des données

3.2.4.1 Méthode d'origine

Dans sa thèse (Lippert, 1998), Lippert propose d'utiliser la technique de la tranche de Fourier (Dudgeon et Mersereau, 1990). Cela vient de la propriété suivante : la transformée de Fourier 2D de la projection orthogonale d'une fonction tridimensionnelle sur un plan selon dans une direction \vec{n} est égale à la tranche centrée orthogonale à la direction \vec{n} de la transformée de Fourier 3D de cette même fonction. La figure 27 illustre le fonctionnement de ce théorème.

L'idée est de Lippert de générer directement la tranche ([3] sur la figure 27, à partir des fonctions analytiques dans le domaine fréquentiel de l'ondelette. Il suffira alors d'y appliquer une transformée de Fourier inverse pour générer la projection désirée. Pour générer la tranche analytique, Lippert (Lippert, 1998) propose d'utiliser les transformées de Fourier continues des fonctions continues d'ondelettes. Pour l'ondelette de Haar, $\Phi(f) = \frac{\sin(\pi f)}{\pi f}$ et $\Psi(f) = 2i \frac{\sin(0.5\pi f)^2}{\pi f}$.

Du fait du rapport dyadique entre les ondelettes à différents niveaux de résolution, seules les empreintes de plus basse définition sont générées (i.e. les empreintes les plus grandes).

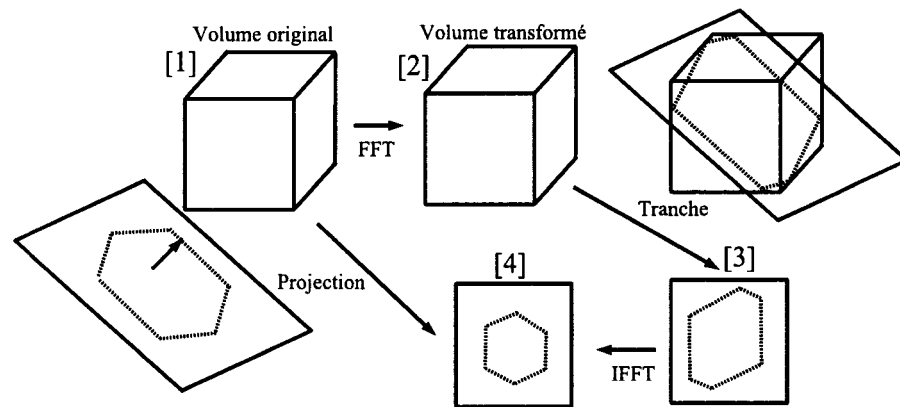


Figure 27 Théorème de la tranche de Fourier. Le volume original [1] subit une FFT pour obtenir le volume transformé [2]. La tranche [3] du volume transformé subit une IFFT pour obtenir la projection [4] du volume original

Les empreintes aux niveaux de résolutions supérieures sont obtenus par un sous échantillonnage par deux des empreintes originales. Afin que ce sous-échantillonnage soit facilement implémentable, les fonctions d'ondelettes sont d'abord centrées et une puissance de deux est choisie pour la taille des ondelettes. Les projections de la figure 28 sont ainsi obtenues.

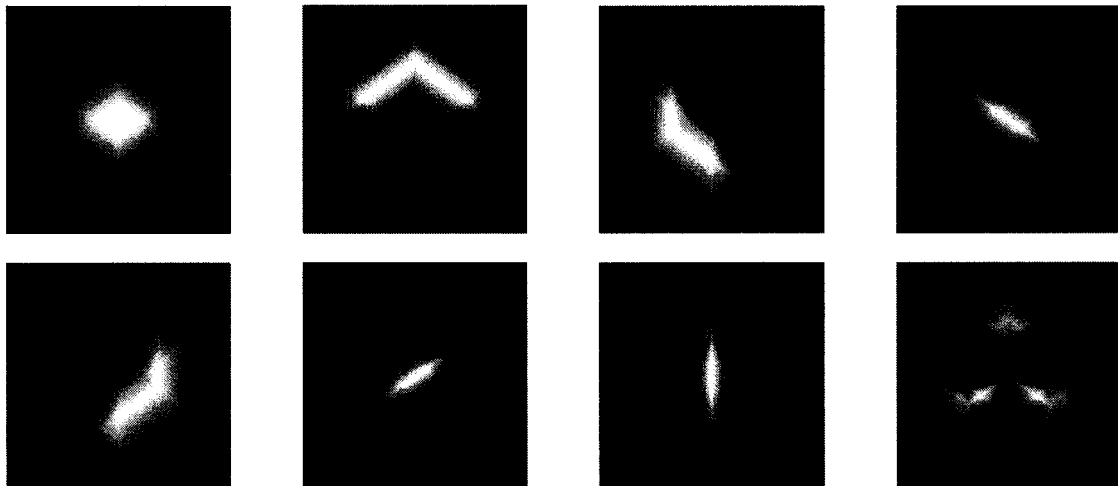


Figure 28 Les 8 projections de l'ondelette de Haar. Chaque projection correspond à un type d'ondelette $\{f_{o,i,0}\}_{i \in [0..7]}$

Si cette technique est utilisée avec l'ondelette de Haar qui a été choisie, certains artefacts apparaissent, notamment dans les directions d'angle multiple de $\frac{\pi}{2}$. En effet, comme l'ondelette de Haar a un support fréquentiel infini et qu'il est impossible de générer une tranche de taille infinie, les bords obtenus ne sont pas francs (figure 29) comme ils devraient l'être puisque la projection d'un cube selon un de ces axes est un carré.

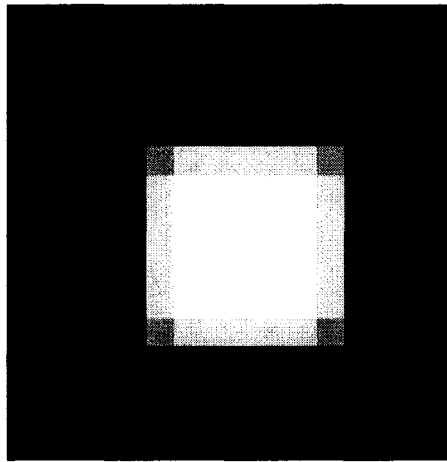


Figure 29 Adoucissement des contours de la projection de l'ondelette de Haar en utilisant le théorème de la tranche de Fourier

Pour éliminer ces artefacts, Westenberg (Westenberg et Roerdink, 2000) propose d'appliquer une fenêtre de Hamming avant la transformée de Fourier inverse. Ceci fonctionne bien. Par contre un autre problème, plus important, apparaît pour les projections des ondelettes de détail. En effet en centrant nos ondelettes, la valeur centrale des projections de ces ondelettes sera toujours nulle et le saut abrupte qui devait être obtenu ne sera plus présent. Cela revient à décaler l'empreinte d'un demi-pixel en haut à gauche et engendre donc une approximation supplémentaire. Horbelt (Horbelt et al., 1999) montre d'ailleurs qu'en utilisant le théorème de la tranche de Fourier, les projections obtenues sont bonnes (en terme d'erreur quadratique moyenne) pour des B-splines cubiques et d'ordre supérieur, et relativement mauvaises pour l'ondelette de Haar.

3.2.4.2 Modifications

Pour y remédier, nous avons pensé modifier légèrement la méthode. Tout se déroule de la même façon sauf que la tranche analytique est générée à partir de la transformée de Fourier discrète de la fonction discrète d'ondelette. Dans le cas de l'ondelette de Haar, pour une tranche générée de taille $N \times N$, on utilise ainsi $\Phi(f) = \frac{\sin(0.5\pi f)}{\sin(\frac{\pi f}{N})}$ et $\Psi(f) = 2i \frac{\sin(0.25\pi f)^2}{\sin(\frac{\pi f}{N})}$. Ceci permet non seulement d'éliminer l'artefact précédent mais aussi de réduire la taille de la tranche à générer. Cette méthode présente le problème que le théorème de la tranche de Fourier n'est plus applicable. Ainsi des artefacts sont observés lorsque l'angle n'est plus un multiple de $\frac{\pi}{2}$, autrement dit la plupart du temps (figure 30).

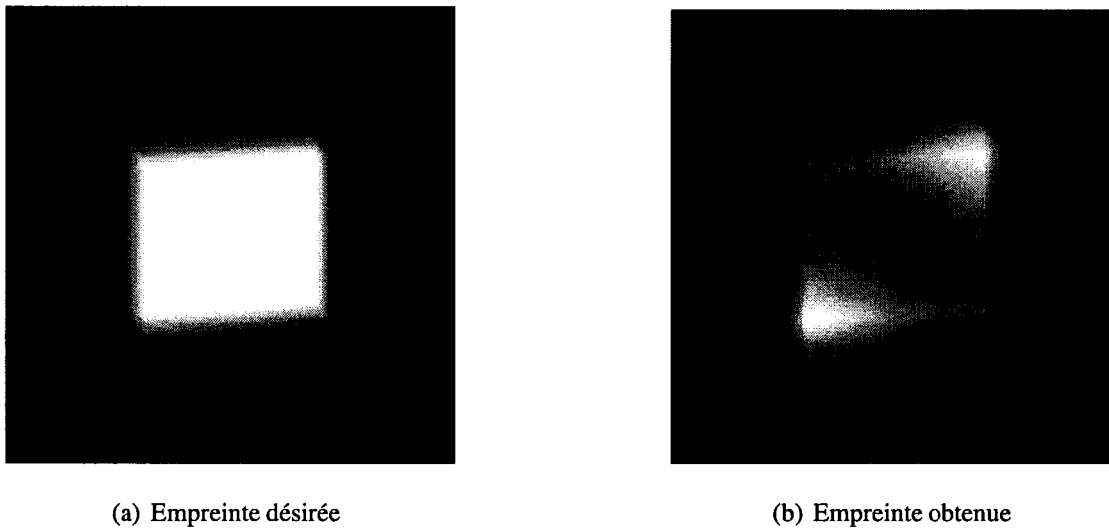


Figure 30 Artefacts de projection en utilisant le théorème de la tranche de Fourier avec la transformée de Fourier discrète de la fonction discrète d'ondelette

Pour cette raison nous avons pris la décision de générer ces empreintes en projetant directement les fonctions tridimensionnelles d'ondelettes. Pour cela, la méthode de projection de voxels a été utilisée au lieu d'une méthode de lancer de rayon. Ce choix a été fait car il paraît plus naturel de projeter de la même manière les empreintes et les fonctions tridimensionnelles. Les fonctions d'ondelettes La complexité passe alors de $O(N^2 \log(N))$

à $O(N^3)$. Cependant l'ondelette de Haar étant très compacte, il s'avère que le temps de génération de ces empreintes bien que plus lent (tableau I), est très faible comparé au temps moyen d'accumulation de ces empreintes. En effet pour un petit volume de taille $256 \times 256 \times 256$, en basse résolution, ce temps moyen obtenu par la même méthode que précédemment est de 39.81 ms.

Tableau I

Comparaison du temps de génération des empreintes. Le temps affiché est une moyenne sur 22 temps de projections d'empreintes d'ondelettes de Haar, selon des angles différents, pour une décomposition de niveau 3, auxquels la plus haute et la plus basse valeur ont été enlevées

	Mode de génération	
	Tranche de Fourier	Projection directe
Durée moyenne (en ms)	1.5	2.8

3.2.5 Projection des empreintes

Une fois que les empreintes ont été calculées, il convient de les projeter au bon endroit et de bonne façon sur l'écran.

3.2.5.1 Matrice de passage

Le système de coordonnées utilisé est celui présenté sur la figure 31.

Pour définir la position des empreintes dans l'espace de visualisation, la matrice de passage de l'équation 3.7 est utilisée. Il est important de calculer la nouvelle coordonnée de profondeur, n , dans le nouvel espace car elle permet de savoir si le voxel considéré se situe devant ou derrière la région d'intérêt.

$$\begin{pmatrix} n \\ u \\ v \end{pmatrix} = \begin{pmatrix} \cos \beta & \sin \beta \cos \alpha & \sin \beta \sin \alpha \\ -\sin \beta & -\cos \beta \cos \alpha & \cos \beta \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3.7)$$

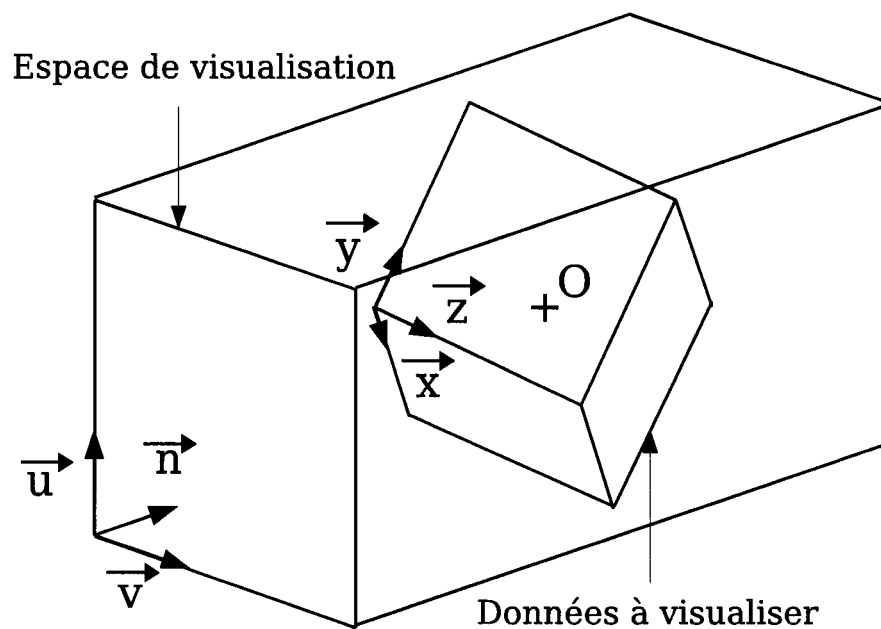


Figure 31 Système de coordonnées. O est l'origine des deux repères. Le repère de l'espace de visualisation est $(O, \vec{u}, \vec{v}, \vec{n})$, celui des données est $(O, \vec{x}, \vec{y}, \vec{z})$

où α et β sont les deux angles de rotation respectivement autour des axes (O, \vec{x}) puis de (O, \vec{z}) . Cette matrice de passage est notée P tandis que la sous-matrice de P permettant de déterminer (u, v) à partir de (x, y, z) est notée Q . Les relations de l'équation 3.8 sont alors obtenues.

$$\begin{aligned}
 P &= \begin{pmatrix} \cos \beta & \sin \beta \cos \alpha & \sin \beta \sin \alpha \\ -\sin \beta & -\cos \beta \cos \alpha & \cos \beta \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \\
 Q &= \begin{pmatrix} -\sin \beta & -\cos \beta \cos \alpha & \cos \beta \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \tag{3.8}
 \end{aligned}$$

$$\begin{aligned}
 (n, u, v)^T &= P \times (x, y, z)^T \\
 (u, v)^T &= Q \times (x, y, z)^T
 \end{aligned}$$

L'algorithme 2 illustre la façon dont la projection s'effectue lorsque le volume a été décomposé 3 fois. Le volume de basse résolution est donc noté $V_{3,0}$.

3.2.5.2 Influence de l'interpolation

En utilisant la matrice de projection, la position centrale de l'empreinte, qui est rarement un entier, est obtenue, une interpolation doit donc être effectuée sur le plan de projection. Une interpolation bilinéaire, qui est rapide et donne une bonne qualité d'image, a été choisie. La figure 32 montre l'importance de cette interpolation.

3.3 Région d'intérêt

Il convient de définir la région d'intérêt. La méthode la plus simple consiste à considérer un sous-volume parallélépipédique rectangle dont les 3 axes principaux sont ceux du volume. Que faire alors des données qui se situent devant ou derrière la région d'intérêt ? Deux choix sont possibles : effectuer le rendu haute résolution de ces données (figure 33(a)) ou les éliminer complètement (figure 33(b)).

Algorithme 2 : Projection des empreintes lorsque le volume est décomposé trois fois

Calcul des empreintes;

Calcul de la région d'intérêt dans le domaine de visualisation;

pour $(i, j, k) \in V_{3,0}$ **faire**

$$(n, u, v)^T = P \times (i, j, k)^T;$$

Ajouter l'empreinte de basse résolution de position originale $4(i, j, k)$ à la position (u, v) ;

si $(u, v, n) \in$ région d'intérêt **alors**

Ajouter les empreintes de détails de niveau 1 de position originale $4(i, j, k)$ à la position (u, v) ;

pour $(i_t, j_t, k_t) \leftarrow (0, 0, 0)$ à $(1, 1, 1)$ **faire**

$$(i_2, j_2, k_2) = (i + \frac{i_t}{2}, j + \frac{j_t}{2}, k + \frac{k_t}{2});$$

$$(u_2, v_2)^T = Q \times (i_2, j_2, k_2)^T;$$

Ajouter les empreintes de détails de niveau 2 de position originale $4(i_2, j_2, k_2)$ à la position (u_2, v_2) ;

pour $(i_t, j_t, k_t) \leftarrow (0, 0, 0)$ à $(1, 1, 1)$ **faire**

$$(i_3, j_3, k_3) = (i_2 + \frac{i_t}{4}, j_2 + \frac{j_t}{4}, k_2 + \frac{k_t}{4});$$

$$(u_3, v_3)^T = Q \times (i_3, j_3, k_3)^T;$$

Ajouter les empreintes de détails de niveau 3 de position originale

$4(i_3, j_3, k_3)$ à la position (u_3, v_3) ;

Le fait de considérer ces deux options nécessite de connaître l'emplacement des données situées devant et derrière la région d'intérêt. Pour faciliter cette tâche, à partir du volume de la région d'intérêt qui est un parallélépipède rectangle dont les 3 axes principaux sont ceux du volume, le parallélépipède rectangle englobant cette région mais dont les 3 axes principaux sont cette fois ceux de l'espace de visualisation est défini. Ceci est illustré sur la figure 34. Après le changement de repère, via la matrice de passage, les coordonnées des empreintes et de la région d'intérêt sont alors tous les deux dans le système de coordonnées de l'espace de visualisation. La région d'intérêt est alors déterminée par six

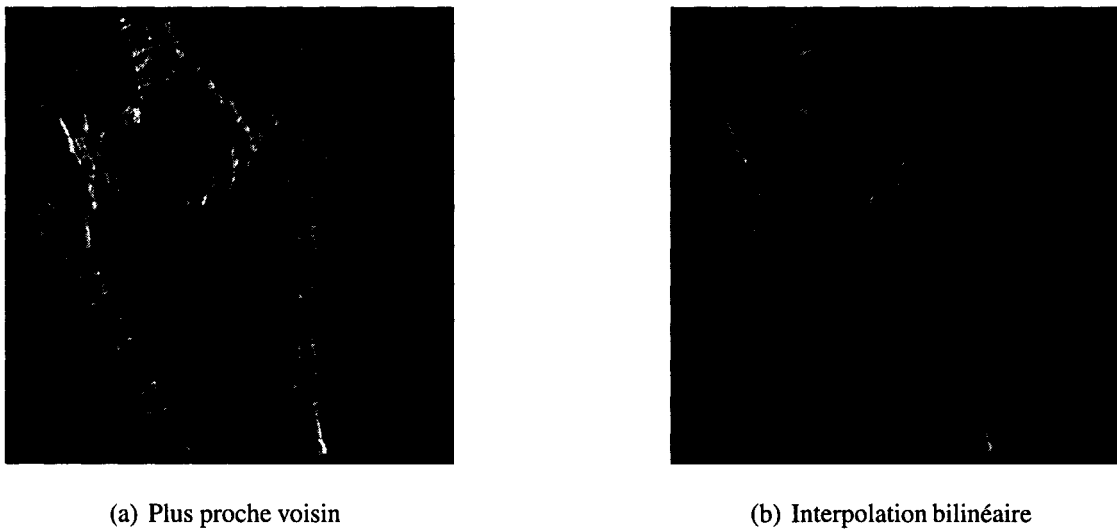


Figure 32 Comparaison du rendu obtenu en utilisant ou non une interpolation bilinéaire sur le plan d'observation

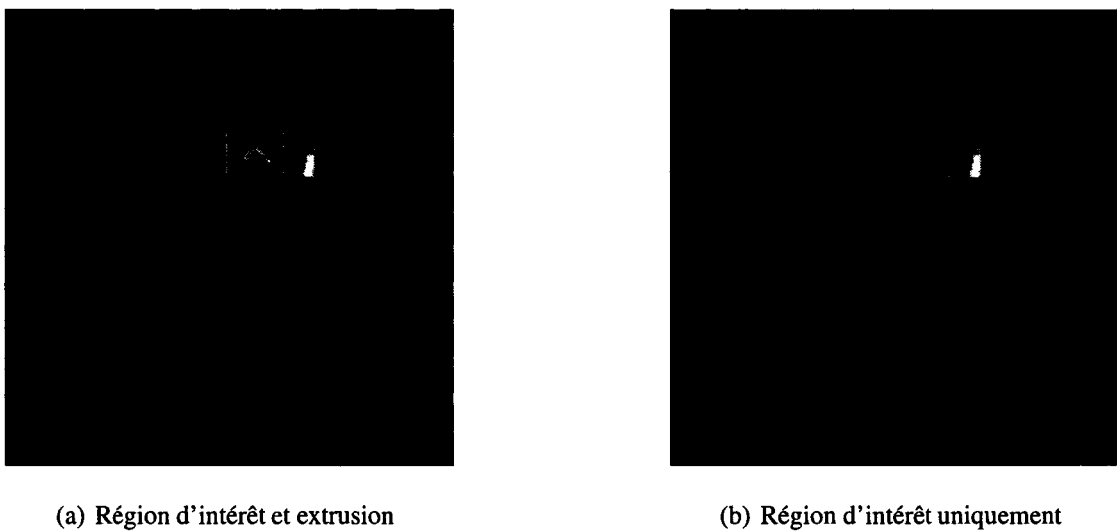


Figure 33 Comparaison du rendu obtenu en affichant ou non les données autour de la région d'intérêt

valeurs : (u_{min}, u_{max}) pour la hauteur, (v_{min}, v_{max}) pour la largeur et (n_{min}, n_{max}) pour la profondeur. Des comparaisons très simples suffisent alors pour savoir si une empreinte est située devant ou derrière la région d'intérêt. Soient (u, v, n) , les coordonnées de la posi-

tion de l'empreinte. L'empreinte est devant ou derrière uniquement si $u_{min} < u < u_{max}$ et $v_{min} < v < v_{max}$ et ($n < n_{min}$ ou $n > n_{max}$).

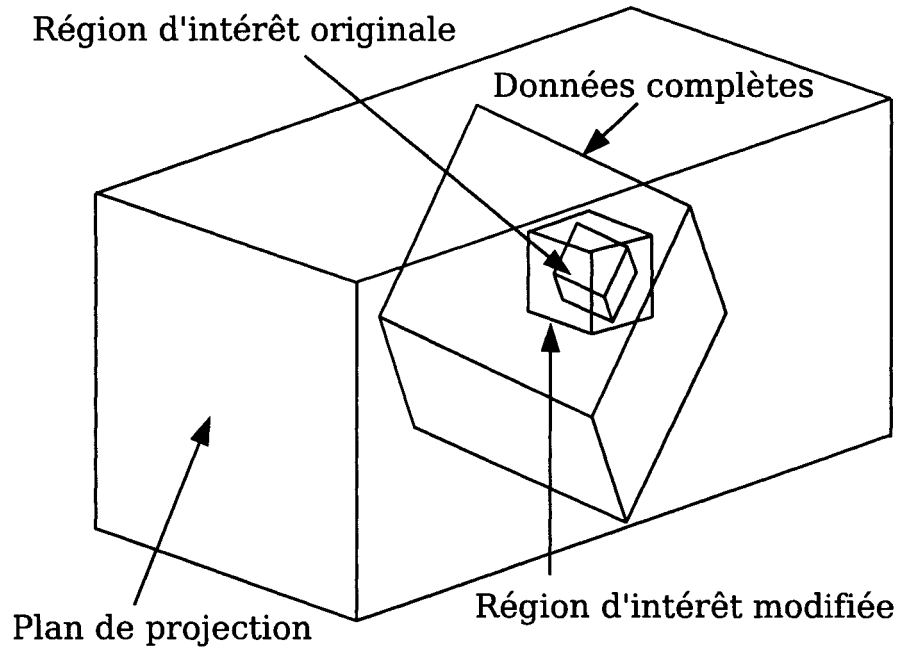


Figure 34 Transformation de la région d'intérêt dans l'espace de visualisation

Une méthode rapide pour calculer les 6 valeurs cités précédemment de ce parallélépipède rectangle est de calculer la projection dans l'espace de visualisation des 8 sommets du parallélépipède rectangle de la région d'intérêt. Ensuite les coordonnées minimum et maximum de ces 8 points selon les 3 axes de visualisation sont déterminées.

3.3.1 Région d'intérêt dans le volume transformé

Pour trouver facilement les coefficients de la région d'intérêt la propriété de localisation de la transformée en ondelettes est utilisée. Ainsi la figure 35 montre les coefficients permettant de reconstruire une région particulière du volume pour un volume décomposé deux fois. Cette relation sera exploitée dans la section suivante pour représenter le volume sous forme d'arbres.

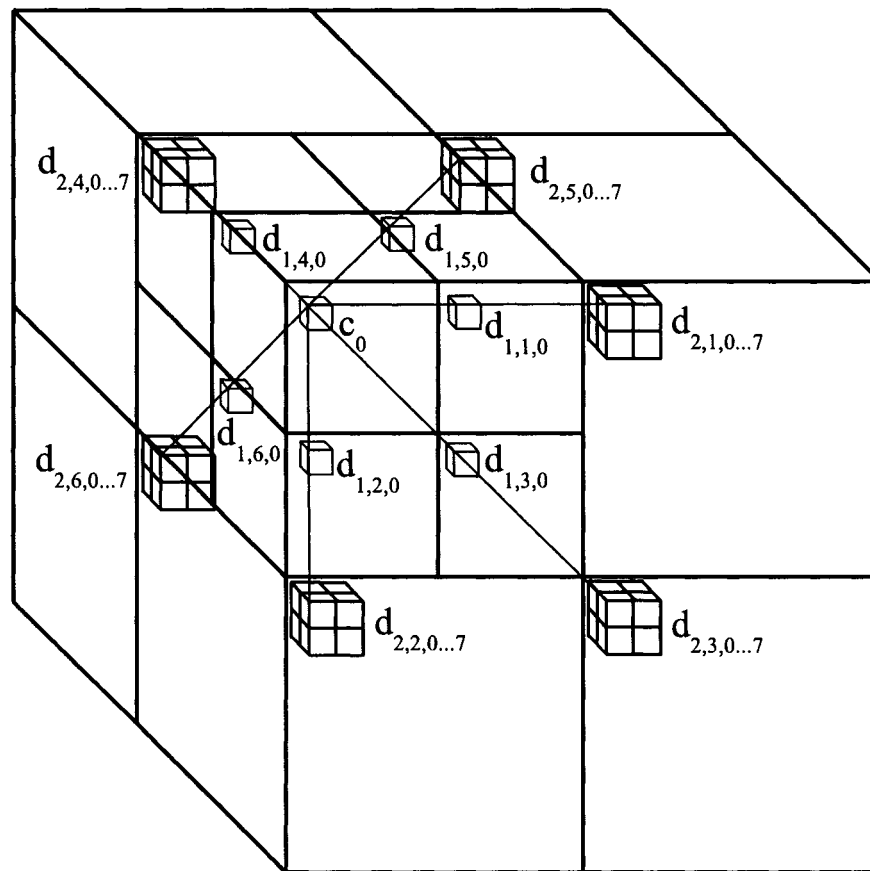


Figure 35 Relation spatiale entre les coefficients. Quand le volume est décomposé deux fois, pour reconstruire une région de $2 \times 2 \times 2$ voxels, il faut un coefficient de basse résolution c_0 , 7 coefficients de détails d'ordre 1 $\{d_{1,i,0}\}_{i \in [1..7]}$, et 7×8 coefficients de détails d'ordre 2 $\{d_{2,i,j}\}_{i \in [1..7], j \in [0..7]}$. Les coefficients $d_{1,7,0}$ et $\{d_{2,7,j}\}_{j \in [0..7]}$, non affichés, sont situés dans les sous volumes non visibles

3.3.2 Structure d'arbres

Cette relation entre les coefficients de chaque sous-volume peut être exploitée afin de représenter différemment nos données. Le volume peut en effet être encodé sous forme d'arbres. Les racines sont les coefficients du sous-volume correspondant à la basse résolution. À chacun de ces coefficients sont associés les coefficients des 7 sous-volumes de même taille. Pour chacun de ces coefficients, les 8 coefficients du sous-volume de même type et de taille 8 fois plus grande sont associés. L'opération est répétée jusqu'à obtention

de tous les coefficients. La structure de la figure 36 est alors obtenue lorsque le volume est décomposé 2 fois.

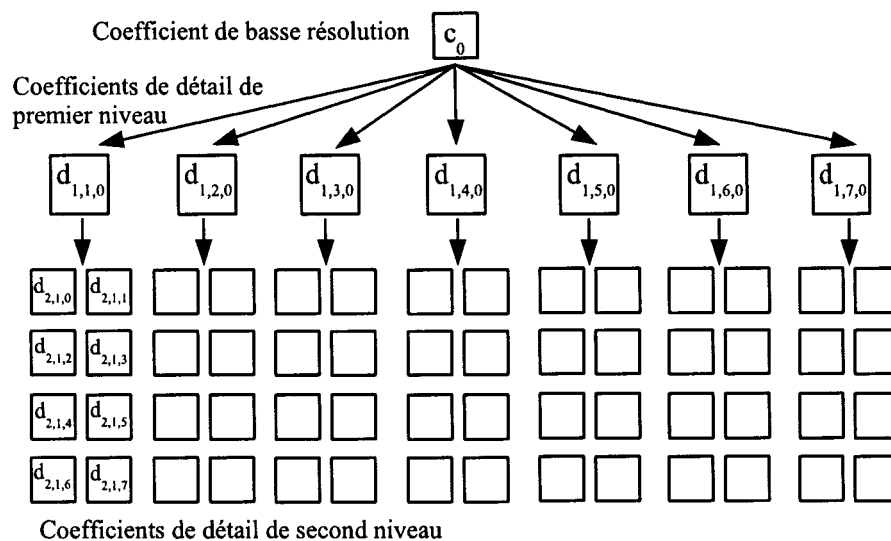


Figure 36 Structure obtenue en utilisant une représentation hiérarchique des coefficients

Comme tous les arbres sont de même profondeur, cette structure peut être enregistrée de façon linéaire comme dans la figure 37 tout en connaissant facilement la position de la racine de chaque arbre.

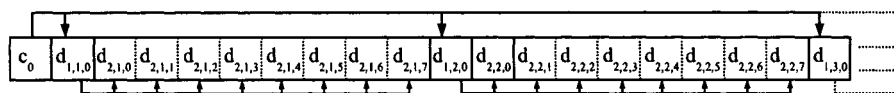


Figure 37 Enregistrement des données de l'arbre de façon linéaire

Cette structure permet en outre de compresser facilement le volume en tronquant les branches lorsque celles-ci ne contiennent que des coefficients nuls. Ceci correspond alors à un espace vide ou à une zone homogène dans le volume original. Dans ce cas une valeur particulière, $ZTREE=0$, est utilisée pour indiquer que le sous-arbre n'est composé que de valeurs nulles. Une valeur $TINY=0.0001$ est utilisée pour indiquer que la valeur du coefficient est nulle mais que le sous-arbre ne l'est pas. Par contre, si cette technique est utilisée

et que les données sont enregistrées sous la même structure que précédemment, il se pose le problème de connaître la position de tête de chaque arbre car ceux-ci ne sont plus de même profondeur et il est très intéressant de pouvoir accéder directement à ces positions sans avoir à traverser tous les sous-arbres. Il suffit alors d'enregistrer ces positions dans une structure séparée (typiquement un vecteur).

La structure sous forme d'arbres tronqués est utilisée dans les techniques de compression comme l'EZW (pour *Embedded Zerotree of Wavelets*) de Shapiro (Shapiro, 1993). Chaque arbre permet de reconstruire une partie différente du volume original. Nous proposons d'ajouter en sus de cette structure la position de la racine de chaque arbre pour permettre un accès rapide à chaque arbre. Ceci permet d'atteindre rapidement et efficacement les différentes régions du volume et est donc adapté à la visualisation d'une région d'intérêt.

3.4 Résumé

En somme, la méthode que nous proposons se base sur :

- des précalculs alliant illumination et ombrage, qui permettent de donner une meilleure impression de volume aux données ;
- une structure de données adaptée au rendu multirésolution avec région d'intérêt ;
- une méthode de rendu proprement dite basée sur la projection d'empreintes d'ondelettes, qui permet de sélectionner une région d'intérêt et d'en effectuer le rendu à pleine résolution, tout en effectuant celui de la région contextuelle à basse résolution.

Le chapitre suivant est dédié aux résultats obtenus en utilisant la méthode proposée.

CHAPITRE 4

RÉSULTATS ET DISCUSSIONS

Ce chapitre présente les différents résultats obtenus en utilisant la méthode décrite précédemment. Les tests ont tous été effectués sur un athlon à 2GHz et 1Go de mémoire vive, sous Linux 2.6.

4.1 Données utilisées

Il n'est pas simple de trouver des données de taille importante librement. Des données présentées sous le format DICOM (pour *Digital Imaging and Communications in Medicine* en anglais), très utilisé en imagerie médicale, ont été utilisées. Ces données proviennent de l'institut Long Beach Memorial et ont été acquises par tomographie hélicoïdale sur un scanner de la compagnie Toshiba. Il s'agit d'un corps (pieds et buste) de 1201 tranches de 512 par 512 pixels (figure 38). Chaque pixel est encodé sur un entier de 16bits. Une grande partie de ces données étant vides, ces données ont été redimensionnées à une taille de $1201 \times 272 \times 472$.

Quelques volumes de plus petite taille sont également disponibles sur Internet¹. Un volume dont la résolution est la même selon les trois axes a été retenu. Il s'agit d'un crâne de 256 tranches de 256 par 256 pixels (figure 39). Chaque pixel est encodé sur un entier de 8bits.

Ce dernier volume étant relativement petit, la plupart des tests ont été effectués sur le premier volume. Chacun de ces volumes a été décomposé trois fois en utilisant l'ondelette de Haar.

¹ <http://www.custard.org/~andrew/visualization/datasets/>

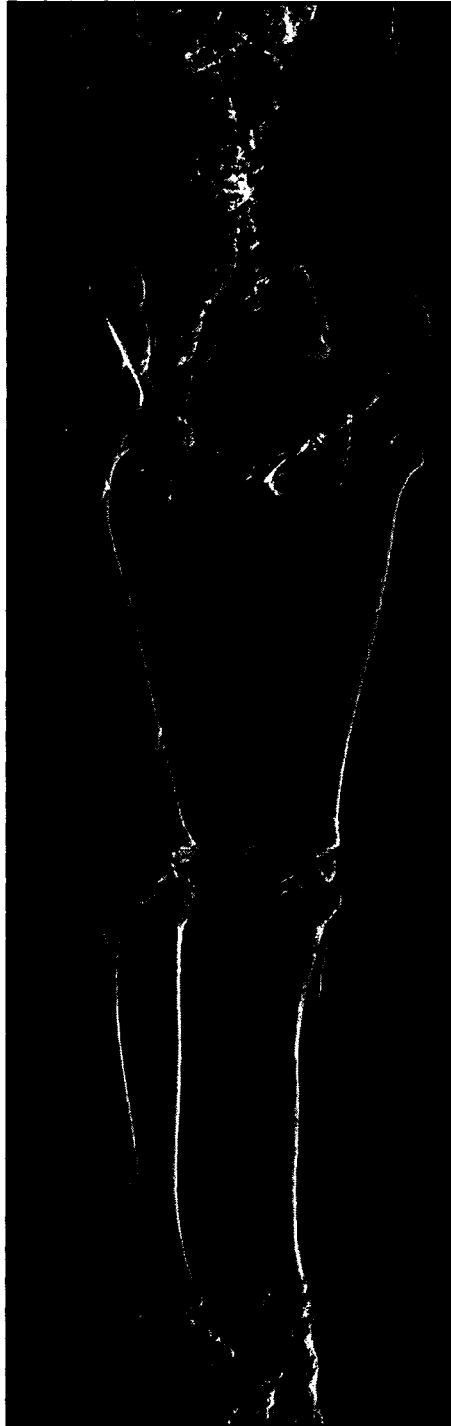


Figure 38 Volume Corps

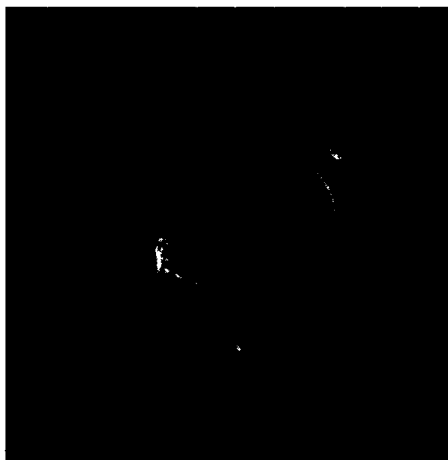


Figure 39 Volume Tête

4.2 Interface utilisateur

Dans le cadre de la visualisation d'images médicales, l'interface utilisateur doit être suffisamment simple pour naviguer autour du volume. L'interface étant plutôt accessoire dans le cadre de ces travaux plus théoriques, une interface minimale a été développée permettant de charger les données, déplacer le volume et la région d'intérêt. Elle est décomposée en deux fenêtres, une pour manipuler le volume, une autre pour configurer le type de rendu.

4.2.1 Chargement des données

La première chose à faire pour utiliser le logiciel est de charger les données en mémoire. Pour cela il suffit de modifier le nom du fichier contenant les données dans le champ prévu à cet effet ([1] sur la figure 40), puis de cliquer sur le bouton de chargement ([5] sur la figure 40). La boîte de dialogue [4] sur la figure 40 permet d'afficher des renseignements techniques, tels que le temps de rendu et le nombre d'empreintes projetées.

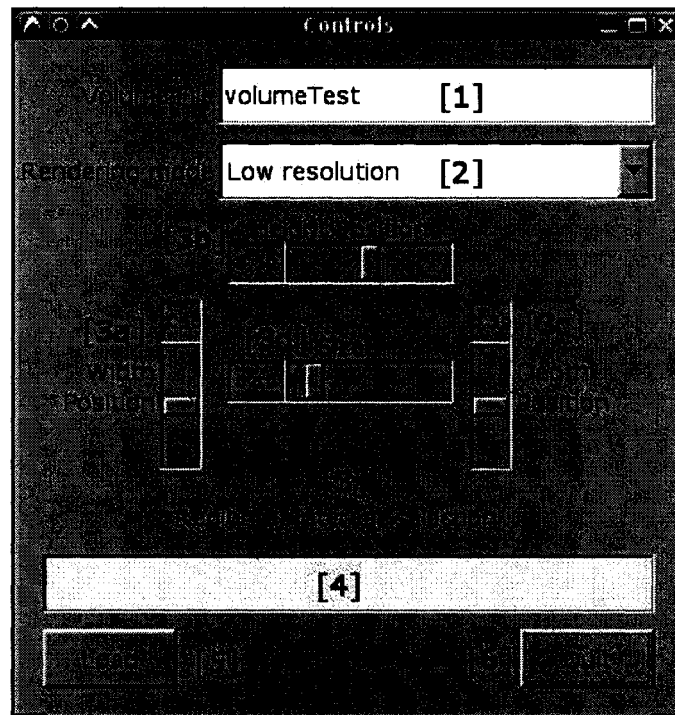


Figure 40 Fenêtre de commande

4.2.2 Modification des modes de rendu

Quatre modes de rendu sont proposés :

- basse résolution ([2a] sur la figure 41) ;
- pleine résolution ([2d] sur la figure 41) ;
- région d'intérêt avec les données devant et derrière ([2b] sur la figure 41),
- région d'intérêt sans ces données ([2c] sur la figure 41).

Lorsque le mode de région d'intérêt est choisi, la région d'intérêt est rendue à pleine résolution tandis que le contexte est rendu à basse résolution. Les deux modes de rendu avec région d'intérêt permettent d'afficher ou d'occulter les données situées devant et derrière.

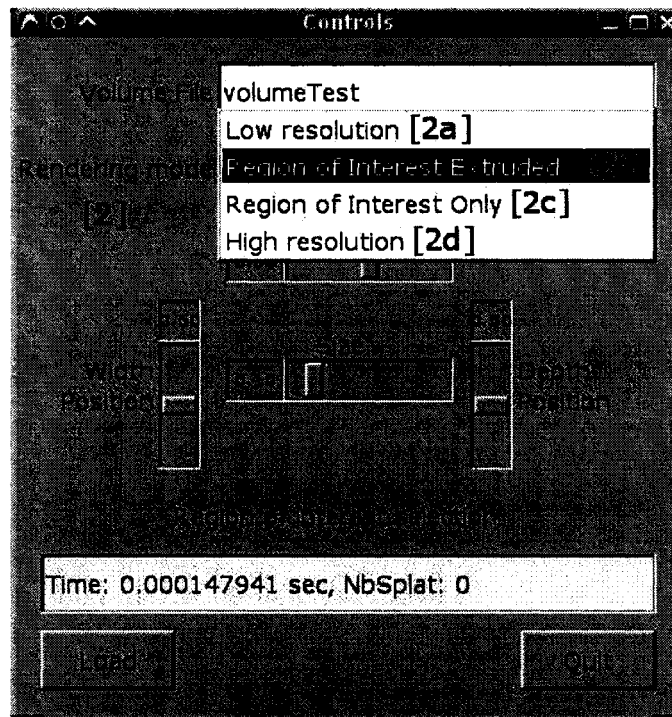


Figure 41 Changement du mode de rendu

4.2.3 Sélection de la région d'intérêt

En ce qui concerne la localisation et le déplacement de la zone d'intérêt, on dispose de 4 barres de navigation dans la fenêtre de commande. Ces barres permettent de déterminer la position centrale de la zone d'intérêt dans l'objet ([3a], [3b] et [3c] sur la figure 40) ainsi que la taille de la région d'intérêt ([3d] sur la figure 40).

4.2.4 Visualisation des données

La fenêtre de visualisation est de taille fixée à 512×512 pixels. Tout d'abord une valeur de saturation peut être choisie afin d'éclaircir ou assombrir l'image ([1] sur la figure 42). Pour naviguer autour du volume, plusieurs outils sont à la disposition de l'utilisateur. La molette et la barre de défilement de gauche ([2] sur la figure 42) permettent respectivement de tourner le volume selon la hauteur du volume, et de le translater verticalement. Enfin

la molette et la barre de défilement en bas ([3] sur la figure 42) permettent de tourner le volume selon la largeur du volume et de le translater horizontalement.

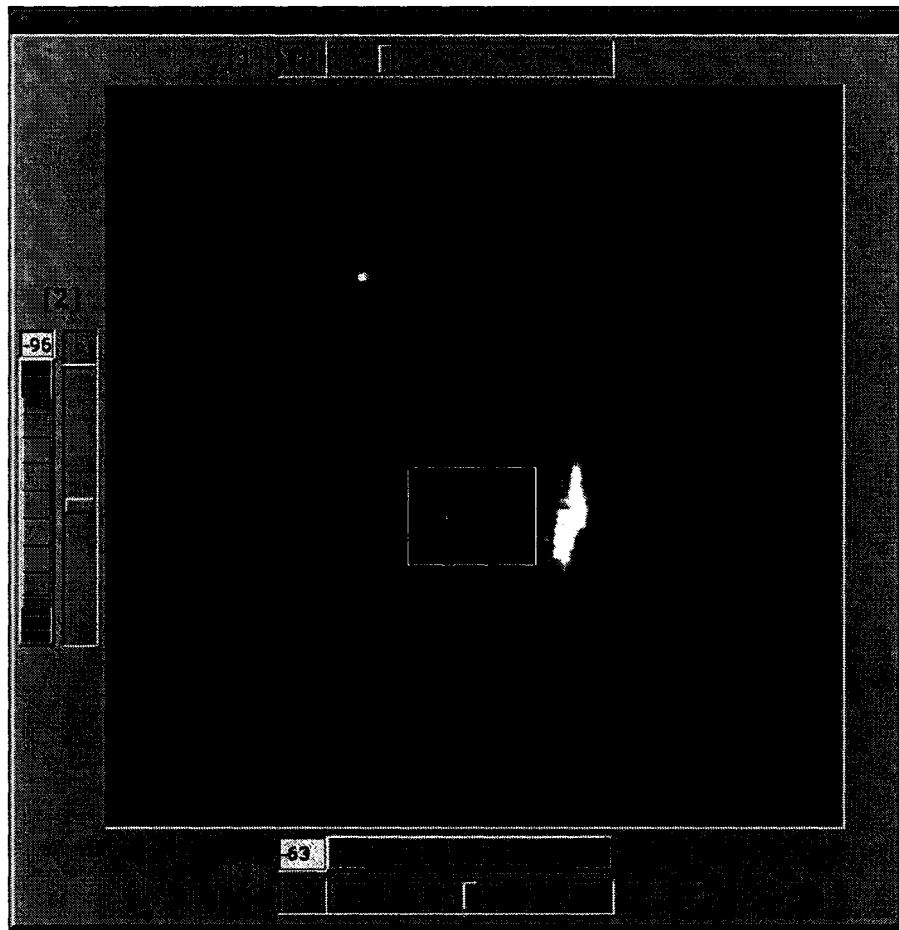


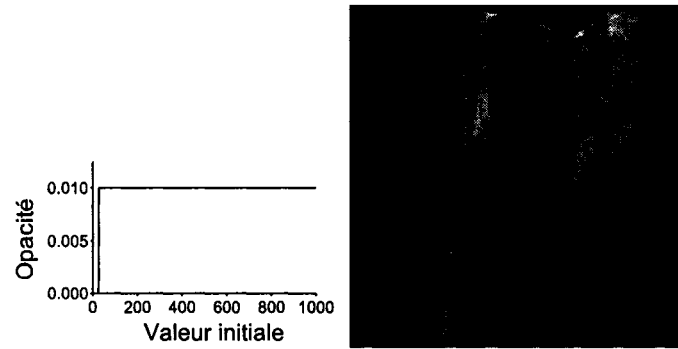
Figure 42 Manipulation du volume

Cette interface nous a permis de manipuler facilement les données ainsi que de nous donner des informations comme le temps de rendu, le nombre de coefficients utilisés ou encore le facteur de normalisation.

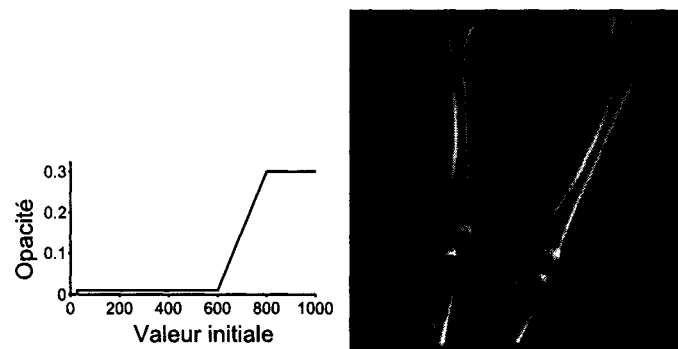
4.3 Influence de la fonction d'opacité sur le rendu final

En jouant sur l'opacité des différents tissus, certains tissus peuvent être atténués voire éliminés (figure 43). Au final, en modifiant les fonctions d'opacité et le modèle d'illumi-

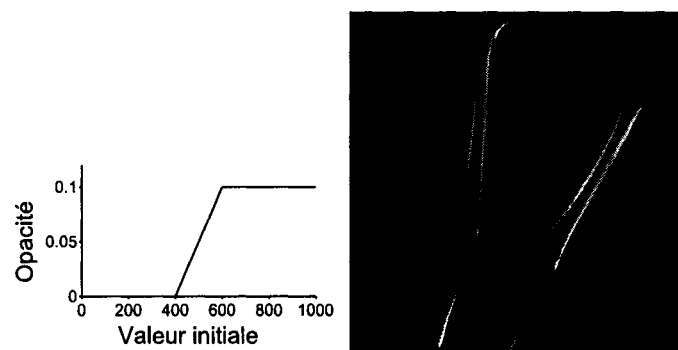
nation, de nombreux rendus différents peuvent être obtenus à partir des mêmes données.



(a) Opacité uniforme



(b) Opacité renforcée sur les os



(c) Opacité ne conservant que le squelette

Figure 43 Effet de l'opacité sur le rendu. A gauche les fonctions d'opacité et à droite le rendu obtenu

4.4 Utilisation d'une post-convolution

Au lieu d'accumuler les empreintes sur le plan de projection, Westenberg (Westenberg et Roerdink, 2000) propose de n'effectuer que la projection des coefficients et d'ensuite utiliser une convolution. Cette technique permet de ne faire qu'une seule interpolation au moment de la projection.

Il s'avère lors de nos tests que l'utilisation d'une post-convolution ralentissait le rendu (tableau II). En fait cela dépend du nombre d'empreintes à ajouter. L'analyse de complexité suivante justifie ce propos.

Tableau II

Comparaison du temps de rendu en basse résolution avec post-convolution ou accumulation d'empreintes

Mode de rendu	Post-convolution	Accumulation d'empreintes
Temps de rendu (en ms) pour le volume Tête	693	51
Temps de rendu (en ms) pour le volume Corps	695	180

Pour simplifier, un volume cubique, des empreintes carrées et un seul sous-volume (celui de basse résolution par exemple) sont considérés. Soit E^2 la taille des empreintes, S^2 la taille de l'écran et N le nombre de coefficients non nuls dans le volume. En projetant les empreintes avec une interpolation bilinéaire, le nombre d'opérations élémentaires est de $N \times E^2 \times 4$. En utilisant une post-convolution, le nombre d'opérations élémentaires devient $4 \times N + S^2 \times E^2$.

Pour que la post-convolution soit intéressante il faut donc que :

$$\begin{aligned}
 N \times E^2 \times 4 &> 4 \times N + S^2 \times E^2, \\
 \text{donc } N &> \frac{S^2 E^2}{4(E^2 - 1)}, \\
 \text{or } E &>> 1, \\
 \text{donc } N &> \frac{S^2}{4}.
 \end{aligned} \tag{4.1}$$

Par exemple, avec un volume de taille 512^3 , une zone d'affichage de taille $S^2 = 512^2$ et un niveau de décomposition de 3, le sous-volume de basse résolution est de taille $V = 64^3$. Le taux de coefficients non nul de ce sous-volume doit donc être de $\frac{N}{V} = \frac{S^2}{4V} = \frac{512 \times 512}{4 \times 64 \times 64 \times 64} = 0.25$. Les deux volumes utilisés présentent un taux de coefficients non nul dans le sous-volume de basse résolution aux alentours de 0.1. Ceci explique donc pourquoi la méthode de post-convolution n'est pas utile dans ce cas.

4.5 Apport de la structure hiérarchique des coefficients

En utilisant une représentation hiérarchique des coefficients et en tronquant les arbres qui n'ont que des fils nuls, gain significatif en occupation mémoire est obtenu. Le tableau III montre le taux de compression obtenu sur les données. Les données sont des nombres flottants et aucune quantification n'a été utilisée ce qui implique une compression sans perte.

4.6 Influence de la taille de la région d'intérêt

La figure 44 montre l'accélération obtenue selon la taille de la région d'intérêt pour le volume Corps. La méthode n'apporte un gain que pour des régions d'intérêts petites, ce qui est une de nos hypothèses de départ.

En regardant d'un peu plus près la zone intéressante (figure 45), c'est-à-dire pour une région d'intérêt de taille inférieure à $20\% \times 20\% \times 20\% = 0.8\%$ du volume total, l'accélé-

Tableau III

Effet de la représentation sous forme d'arbres tronqués du volume. Pour le volume Corps, la fonction d'opacité est celle utilisée à la figure 43(b). Pour le volume Tête, la fonction d'opacité est uniforme

Volume considéré	Corps	Tête
Taille originale	588 Mo	65 Mo
Taille avec structure d'arbre	126 Mo	6.8 Mo
Gain	4.67	9.55

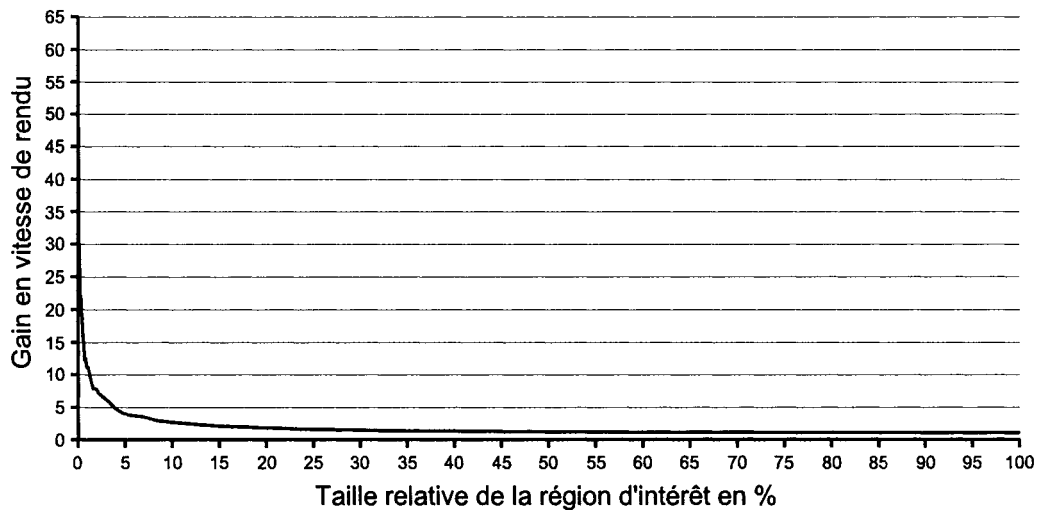


Figure 44 Influence de la taille de la région d'intérêt. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer un rendu avec région d'intérêt et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille totale du volume. La fonction d'opacité utilisée est celle de la figure 43(c)

ration apportée par la considération d'une zone d'intérêt est significative. Pour une région d'intérêt de $10\% \times 10\% \times 10\% = 0.1\%$ du volume, une accélération d'un facteur 30 est ainsi obtenue.

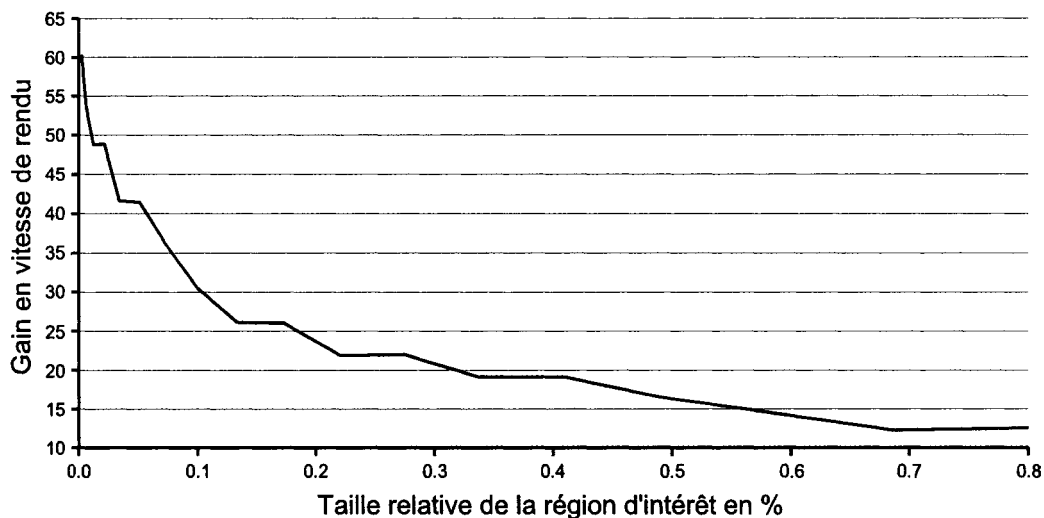


Figure 45 Influence de la taille de la région d'intérêt sur la rapidité du rendu lorsque la région est plus petite. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer un rendu avec région d'intérêt et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille totale du volume. La fonction d'opacité utilisée est celle de la figure 43(c)

Enfin, d'autres résultats, obtenus sur les larges données présentées à la section 4.1 en considérant une opacité renforcée sur les os (voir figure 43(b)), sont présentés. Le temps de rendu moyen est de 19.21 secondes en haute résolution et de 0.43 secondes en basse résolution. Le fait d'utiliser des régions d'intérêt permet bien d'accélérer le rendu. Lorsque la région d'intérêt est relativement grande (figures 46), le gain, bien que significatif, n'est pas suffisant pour permettre une navigation agréable. Avec une région d'intérêt plus petite (figure 47), le gain est nettement plus intéressant.

En ne conservant que les os (voir figure 43(c)), et en se focalisant sur un genou, les résultats obtenus sont encore meilleurs (figure 48), puisque toute la partie des tissus mous est éliminée. Ainsi la pleine résolution est rendue en 11.2 secondes et la basse résolution en 0.08 secondes.

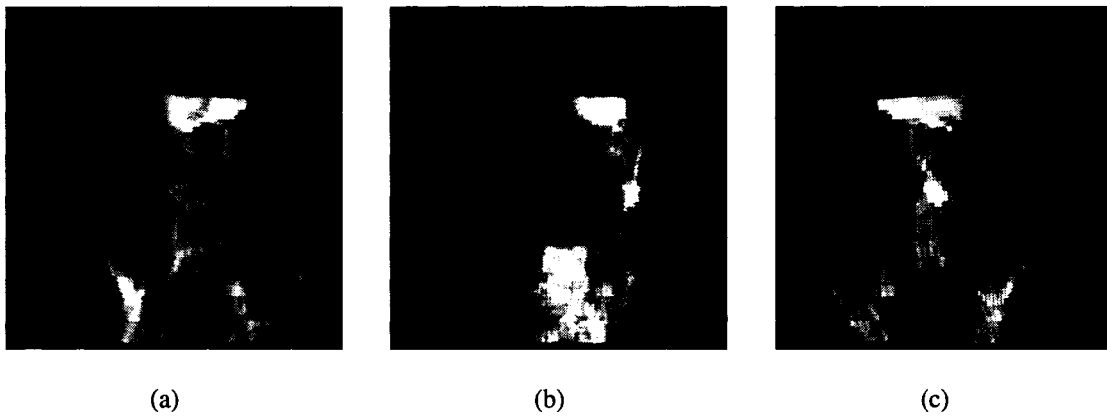


Figure 46 Région d'intérêt centrée sur le foie. La région d'intérêt est de taille $123 \times 123 \times 123$ voxels, soit 1.21% du volume original. Le temps de rendu moyen est de 1.88 seconde, soit un gain en vitesse d'un facteur 10.6 par rapport au rendu pleine résolution

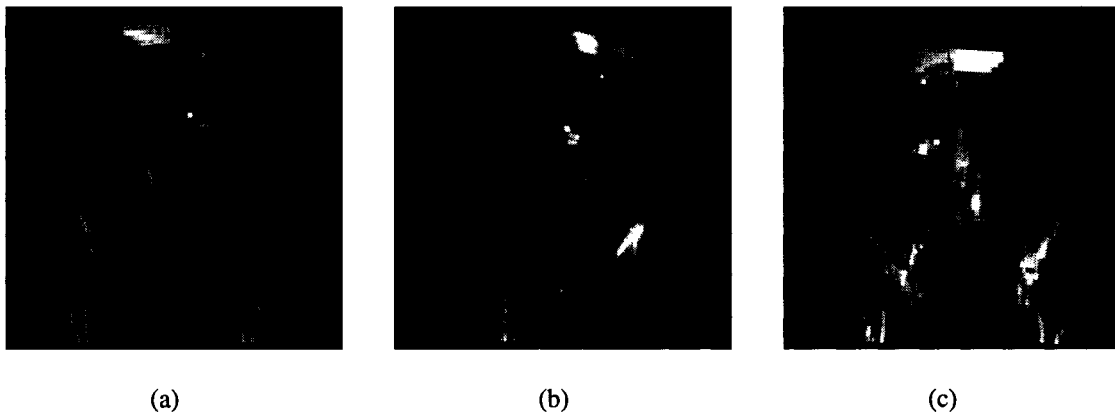


Figure 47 Région d'intérêt centrée sur une partie d'une artère. La région d'intérêt est de taille $52 \times 52 \times 52$ voxels, soit 0.09% du volume original. Le temps de rendu moyen est de 0.65 secondes, soit un gain en vitesse d'un facteur 29.6 par rapport au rendu pleine résolution

Le tableau IV récapitule l'apport de l'utilisation d'une région d'intérêt sur le volume Corps. En particulier, la taille de la région d'intérêt n'est pas le seul facteur influant sur le gain en vitesse en rendu. La fonction d'opacité et la région d'intérêt choisie ont également une influence. En effet un gain en vitesse très proche est obtenu pour une taille de région d'intérêt et une fonction d'opacité différentes (figure 47 et 48). Ceci est normal

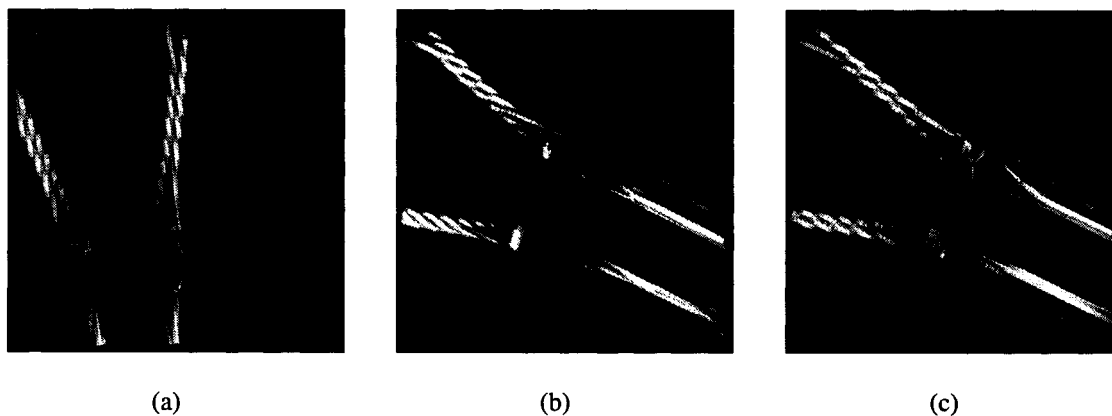


Figure 48 Région d'intérêt centrée sur un genou. La région d'intérêt est de taille $75 \times 75 \times 75$ voxels, soit 0.27% du volume original. Le temps de rendu moyen est de 0.36 secondes, soit un gain en vitesse d'un facteur 31.1 par rapport au rendu pleine résolution

étant donné que la complexité de l'algorithme de projection d'empreintes est directement fonction du nombre de coefficients d'ondelettes non nul. En éliminant les tissus mous ou en changeant de région d'intérêt ce nombre de coefficients varie.

Tableau IV

Accélération apportée par l'utilisation d'une région d'intérêt. Le temps de rendu est exprimé en secondes. Le gain en vitesse de rendu est le rapport entre le temps mis pour effectuer le rendu courant et celui pour un rendu haute résolution. La taille relative de la région d'intérêt est le rapport entre la taille de cette dernière avec la taille du volume total, exprimé en pourcentage

	Haute résolution		Région d'intérêt			Basse résolution	
	Temps (s)	Gain	Taille relative (%)	Temps (s)	Gain	Temps (s)	Gain
Figure 46	19.21	1.0	1.21	1.88	10.6	0.43	44.67
Figure 47	19.21	1.0	0.09	0.65	29.6	0.43	44.67
Figure 48	11.2	1.0	0.27	0.36	31.1	0.08	140

4.7 Résumé

En utilisant la méthode décrite dans le chapitre précédent, le gain en vitesse de rendu apporté par l'utilisation d'une région d'intérêt est vraiment significatif pour des régions couvrant environ 0.1% du volume original.

CONCLUSION

Au travers de ce mémoire, une technique de rendu, alliant à la fois un rendu haute définition dans une région d'intérêt et une manipulation rapide des données, a été proposée. Cette méthode exploite les propriétés de multirésolution et de localisation des ondelettes. La multirésolution est utilisée pour effectuer des rendus de différentes résolutions, tandis que la localisation est utilisée pour identifier, sélectionner et modifier une région d'intérêt parmi les données. Le rendu s'effectue en utilisant la technique de projection d'empreintes d'ondelettes afin d'éviter une reconstruction du volume 3D. Nous avons proposé d'utiliser des précalculs sur les données afin d'obtenir différents modes de visualisation, notamment en définissant une fonction d'opacité. Ces différents modes permettent par exemple de n'afficher que les données utiles. Le travail a porté non seulement sur la méthode de rendu proprement dite mais également sur la représentation des données et la façon de les manipuler. Nous avons ainsi proposé une interface utilisateur afin de manipuler les données de façon intuitive (section 4.2).

Retour sur les résultats

L'utilisation de la technique de projection d'empreintes d'ondelettes limite la qualité du rendu puisqu'il est impossible d'effectuer une intégration de l'opacité des tissus selon la direction d'observation. Toutefois, des précalculs d'illumination et d'ombrage peuvent grandement améliorer la qualité de l'image obtenue et s'avérer suffisante en vue d'effectuer un diagnostic.

Du point de vue vitesse, les résultats obtenus sur nos données montrent une accélération d'un facteur 30 sur le temps de rendu lorsqu'une petite région d'intérêt est utilisée. Cela permet à l'utilisateur de modifier le point de vue sans avoir à changer de résolution. Cette région d'intérêt est en outre modifiable directement via une interface utilisateur. Ces ré-

sultats ont fait l'objet d'une présentation à une conférence internationale (Piccand et al., 2005).

La structure de données proposée, constituée d'arbres tronqués et d'une table d'accès aux racines de l'arbre, exploite pleinement la propriété de localisation des ondelettes afin d'accéder rapidement aux différentes régions des données. Elle permet également une compression sans perte d'un facteur supérieur à 3.

L'ensemble de ces résultats fera l'objet d'un article (Piccand et al., 2006).

Améliorations possibles

Les résultats obtenus sont encourageants, d'autant que le rendu effectué est uniquement logiciel. En effet, la technique de projection des empreintes peut tirer profit des capacités du matériel graphique. Les empreintes pourraient être chargées comme textures et accumulées via le matériel graphique. D'autre part, la génération des empreintes pourrait être effectuée au travers du matériel graphique en utilisant des textures tridimensionnelles. La pleine exploitation du matériel graphique est ainsi envisagée pour les études futures.

La méthode se basant sur la projection d'empreintes d'ondelettes, devrait permettre un rendu de type progressif. Lorsque l'utilisateur ne déplace pas le point de vue, il pourrait être intéressant d'ajouter des détails dans la région contextuelle.

D'autre part, la manipulation des données est également un point important. Il serait bon par exemple de pouvoir définir plusieurs régions d'intérêt et de déplacer encore plus facilement ces données en utilisant le glisser-déposer de la souris. L'utilisation de vues multiples, selon les trois axes majeurs de l'espace de visualisation permettrait également de définir plus facilement la région d'intérêt. L'interface proposée limite l'utilisateur à une région d'intérêt cubique mais la méthode permet d'utiliser n'importe quel parallélépipède rectangle. L'ergonomie est réellement un point important lors de la manipulation des don-

nées. Pour l'améliorer, un partenariat très fort entre le chercheur et l'utilisateur final serait très utile.

Enfin, le rendu final pourrait être amélioré par exemple en normalisant différemment les données du contexte et de la zone d'intérêt ou en incorporant un modèle plus sophistiqué d'illumination tenant par exemple compte de la distance des données au plan de visualisation.

ANNEXE 1

Implémentation

IMPLÉMENTATION

Cette annexe a pour but de présenter les détails d'implémentation de notre méthode. En effet plusieurs choix ont été pris à propos de l'organisation d'un logiciel permettant de manipuler les données.

1 Choix du langage de développement

Notre objectif principal était de développer un logiciel permettant de faire la démonstration de notre méthode proposée. Nous voulions que ce logiciel puisse fonctionner au moins sous les plateformes Unix et Windows. D'autre part, nous avons décidé de ne pas exploiter les optimisations propre au matériel dans un souci à nouveau d'indépendance vis à vis de la plateforme.

La première idée que nous avons eue était d'utiliser et modifier le plugin VolumeJ pour ImageJ. Ce plugin implémente en effet différentes techniques classiques de rendu (iso-surface, lancé de rayon). D'autre part, ce logiciel est programmé en Java et de ce fait est exploitable sur de nombreuses plateformes. Nous avons ainsi validé notre méthode via ce plugin et les modifications que nous avons apporté. Malheureusement notre objectif était de pouvoir effectuer le rendu sur des volumes de grande taille et ce plugin est assez limité au niveau de la structure du volume à utiliser et, surtout, nécessite que le volume soit chargé dans ImageJ sous une certaine forme avant d'être copié sous une autre forme et exploité dans VolumeJ. La gestion de la mémoire posait alors problème. D'autre part il y avait des fuites de mémoire lorsque l'utilisateur désirait changer de volume.

Pour ces raisons, nous nous sommes tournés vers le développement de notre application personnelle. Notre objectif était toujours de développer une application multiplateforme. Étant plus à l'aise avec la programmation en C++, nous avons choisi ce langage. Plus particulièrement, nous avons opté pour le compilateur GCC qui peut fonctionner sous

windows via MINGW. Une interface graphique étant plus agréable qu'une ligne de commande, nous avons dû également choisir une librairie graphique. Comme nous ne nécessitions pas beaucoup de fonctionnalités, nous avons opté pour une librairie multiplateforme minimaliste. Fast Light ToolKit (FLTK) a répondu à nos attentes.

Nous avons séparé la partie rendu proprement dite, qui implémente la méthode de rendu et l'interface utilisateur.

2 Vue générale

La figure 49 montre la représentation UML des différentes classes.

3 Programme principal

Le programme principal a été découpé en plusieurs classes.

- La classe abstraite **VolumeView** permet de gérer le rendu global d'un Volume.
- Les classes **VolumeViewTree** et **VolumeViewDecompose** sont des spécialisations de la classe précédente.
- La classe **Volume** permet de gérer un volume.
- Les classes **Transformer** et **Lighter** permettent d'effectuer les précalculs sur un objet de type **Volume**.
- La classe **Footprint** permet de générer les empreintes des ondelettes.
- La classe abstraite **Wavelet** permet de définir ce que nous avons besoin de connaître pour exploiter un type d'ondelette.
- La classe **Haar** permet de définir l'ondelette de Haar.
- Les classes **VolumeDecompose** et **VolumeTree** permettent de définir deux types de représentation d'un volume qui a subi une décomposition par ondelettes. **VolumeDecompose** représente le volume comme un ensemble de sous volumes correspon-

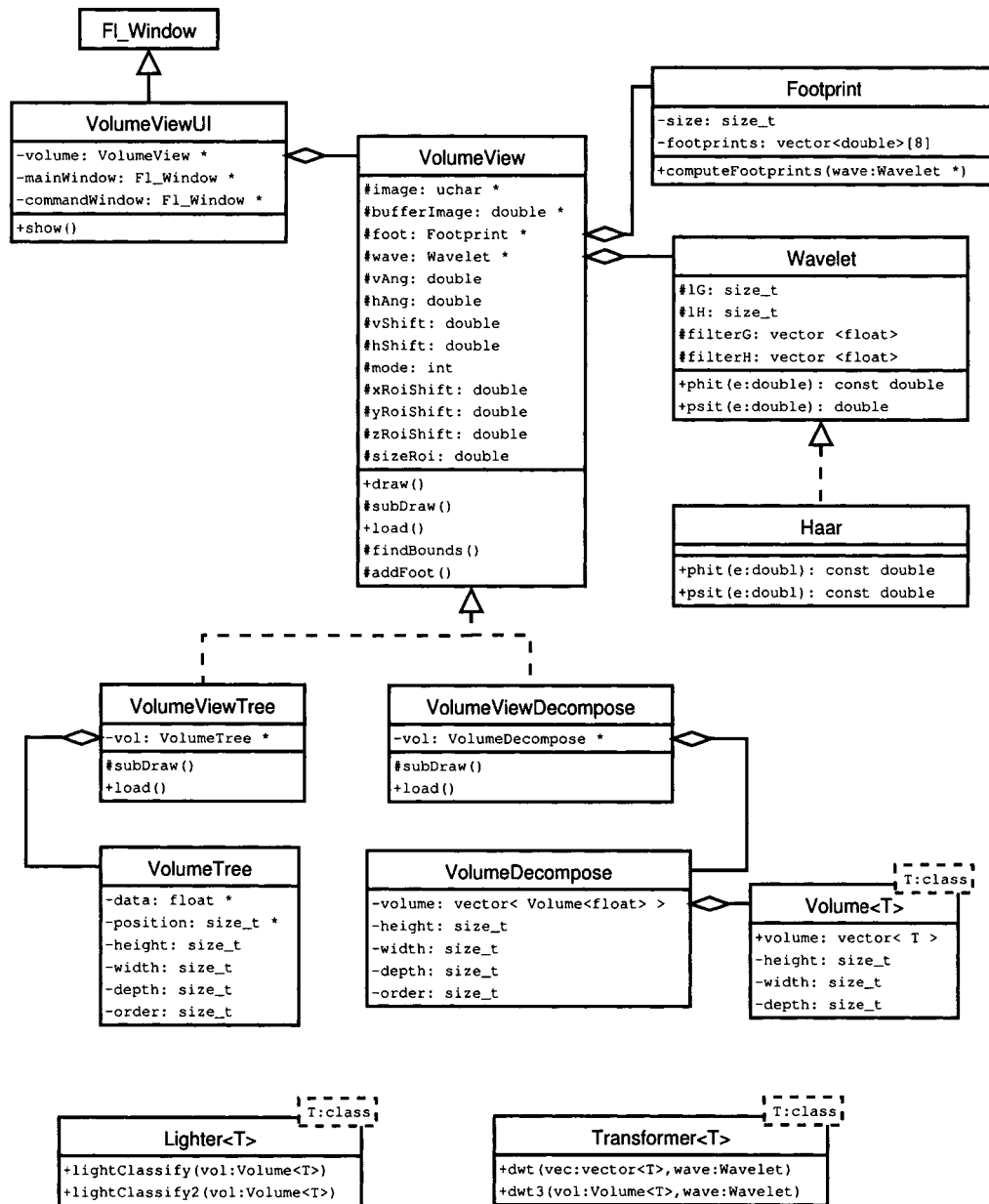


Figure 49 Diagramme des classes du logiciel développé. Seuls les attributs, méthodes et arguments les plus importants sont représentés

tant à chaque sous bande. **VolumeTree** représente le volume comme un ensemble d'arbres comme décrit à la section 3.3.2.

3.1 La classe template **Volume**

Cette classe permet de gérer les données d'un volume. Pour cela il est nécessaire de définir la hauteur (**height**), la largeur (**width**) et la profondeur (**depth**) du volume.

Ensuite, les données sont stockées dans le vecteur **volume**. Par ailleurs nous avons défini une méthode de lecture **read()** qui permet de lire un volume dont les données sont enregistrées de façon brute (selon la hauteur, la largeur, puis la profondeur). Ainsi la donnée de coordonnées (i,j,k) se trouve à la position $volume[i + height * (j + width * k)]$. Ceci est transparent du point de vue de l'utilisateur, étant donné que il doit utiliser les accesseurs appropriés pour modifier ou obtenir la valeur d'un élément du volume.

3.2 Les classes template **Transformer** et **Lighter**

Ces classes permettent de réaliser les précalculs sur un volume. La classe **Transformer** permet d'effectuer la transformée en ondelettes via la méthode **dwt3()**. La classe **Lighter** permet de faire le calcul d'illumination via les méthodes **lightClassify()** si l'opacité n'est pas prise en compte, et **lightClassify2()** si elle est prise en compte.

3.3 La classe **VolumeDecompose**

Cette classe permet de gérer un volume qui a subi une décomposition en ondelettes sous forme de sous volumes. La taille du volume original (**height**, **width** et **depth**) ainsi que son niveau de décomposition (**order**) doivent être connues.

Les données sont ensuite stockées sous forme de sous volume dans un vecteur de **Volume<float> volume** de taille $7 \times order + 1$. En effet, il contient le sous-volume comprenant les coefficients de basse résolution, plus les 7 types de sous volumes comprenant les coefficients de détail selon les 7 directions possibles. De plus il y a autant de sous volumes de détails que le niveau de décomposition du volume. Ces différents volumes sont

accessibles via l'accesseur **getSubvolume()**. Utilisé sans paramètre le volume de basse résolution est obtenu, sinon il faut préciser le type de détails ainsi que le niveau de détail désiré.

3.4 La classe **VolumeTree**

Cette classe permet de gérer un volume qui a subi une décomposition en ondelettes sous forme de sous arbres. Outre les attributs de taille et de niveau de décomposition, il y a un tableau, **data**, contenant les sous arbres et un autre, **position** permettant de retrouver la position de l'origine de chaque sous arbre. Ce dernier tableau est utile pour éliminer les branches vides des arbres et accéder rapidement aux racines de chaque arbre.

Étant donné la structure des données utilisée, il est peu efficace d'accéder aux sous volumes de la même façon que dans **VolumeDecompose**. Les données sont donc directement accessibles.

3.5 La classe **Footprint**

Cette classe permet de générer les empreintes des ondelettes. Les empreintes sont stockés dans un tableau de 8 vecteurs de doubles, **footprints**. Chaque vecteur contient une empreinte selon une sous bande. Les empreintes sont carrées et de taille **size**.

La méthode principale pour générer les empreintes est **computeFootprint()**. Elle prend en compte une ondelette, **Wavelet**, ainsi que les sinus et cosinus des deux angles d'observation. Il est inutile de préciser l'ordre de l'ondelette car celui-ci est déterminé par sa taille.

3.6 La classe abstraite **Wavelet**

Cette classe a été créée afin de spécifier ce qui doit être connu à propos d'une ondelette particulière. Il est nécessaire de connaître les coefficients de ses deux filtres, **filterG** et

filterH, afin de décomposer le volume. D'autre part, il est nécessaire de connaître la fonction analytique des ondelettes, dans le domaine fréquentiel, **phi()** et **psi()**, si le théorème de la transformée de Fourier tranchée est utilisé, ou dans le domaine temporel, **phit()** et **psit()** si la projection directe de l'ondelette est utilisée. Les coefficients des filtres **filterG** et **filterH** sont enregistré dans des vecteurs. Leur taille, **lengthG** et **lengthH**, ainsi que leur ordre, **orderG** et **orderH**, sont également nécessaires lors de la décomposition.

3.6.1 La classe Haar

La classe **Haar** permet de définir l'ondelette de Haar. Sont définis ainsi :

- **filterG** : $[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$,
- **filterH** : $[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]$,
- **phi()** : $\phi(f) = \text{sinc}(f)$,
- **psi()** : $\psi(f) = 2i \sin(\frac{\pi}{2}f) \text{sinc}(\frac{f}{2})$,
- **phit()** : $\phi(t) = 1$ pour $0 \leq t < 1$, $\phi(t) = 0$ ailleurs,
- **psit()** : $\psi(t) = 1$ pour $0 \leq t < \frac{1}{2}$, $\psi(t) = -1$ pour $\frac{1}{2} \leq t < 1$, $\psi(t) = 0$ ailleurs.

3.7 La classe abstraite VolumeView

Il s'agit de la classe principale permettant d'effectuer le rendu de nos données. Les calculs sont effectués temporairement en double sur le tableau dynamique **imageBuffer**. L'image finale **image** est obtenue en normalisant le tableau précédant entre 0 et 256.

Elle possède une méthode **draw()** qui affiche le résultat du rendu et est appelée par l'interface utilisateur. Cette méthode fait appel à la méthode virtuelle **subDraw()** qui doit être définie par les classes spécialisées. Les méthodes protégées **findbounds()**, **addFoot()** et **normalize()** permettent respectivement de déterminer la zone d'intérêt dans l'espace de visualisation, d'ajouter une empreinte à une position précise et de normaliser l'image finale. Les empreintes sont quant à elles générées en faisant appel à la classe **Footprint**.

La méthode virtuelle **load()**, quant à elle, permet de charger les données à visualiser.

Différents accesseurs sont disponibles afin de modifier les paramètres de rendu : angles de rotation (**vAng** et **hAng**), translations verticale (**vShift**) et horizontale (**hShift**), types de rendu (**mode**) et paramètres pour la définition de la zone d'intérêt (position du centre : **xRoiShift**, **yRoiShift** et **zRoiShift** ; taille : **sizeRoi**).

3.7.1 Les classes **VolumeViewTree** et **VolumeViewDecompose**

Ces deux classes, spécialisations de la classe **VolumeView** ne diffèrent que par la représentation des données qu'elles utilisent. Tandis que la classe **VolumeViewTree** utilise un objet de type **VolumeTree**, la classe **VolumeViewDecompose** utilise un objet de type **VolumeDecompose**.

La méthode **subDraw()** permet d'effectuer le rendu proprement dit, décrit dans l'algorithme 2.

4 Interface utilisateur

L'interface utilisateur est gérée par la classe **VolumeViewUI**. Cette interface est découpée en deux fenêtres **Fl_Window** : une de commande (**commandWindow**) et une de visualisation (**mainWindow**).

4.1 Fenêtre de commande

La fenêtre de commande sert à déterminer les données à traiter, à déterminer le niveau de résolution désiré ainsi qu'à la sélection de la zone d'intérêt.

4.1.1 Sélection de la zone d'intérêt

La zone d'intérêt est un cube dans le domaine du volume. La position de son centre est déterminée avec trois barres de défilement de type **Fl_Value_Slider** permettant la translation de cette position selon les trois axes majeurs du volume. La taille de la région est également déterminée à partir d'une barre de défilement de type **Fl_Value_Slider**.

4.1.2 Choix du mode de rendu

Le mode de rendu est déterminé grâce à un menu déroulant de type **Fl_Choice**.

4.1.3 Chargement du volume

Le nom du volume est entré dans une boîte de dialogue de type **Fl_Input**. Il est chargé en cliquant sur le bouton de type **Fl_Button**.

4.2 Fenêtre de visualisation

La fenêtre de visualisation est composée de 4 parties distinctes : la zone de visualisation, deux barres de translation, deux barres de rotation et une barre d'éclaircissement.

4.2.1 Zone de visualisation

La zone de visualisation qui est de taille fixe est de type **VolumeView** puisque c'est l'endroit où les données sont affichées.

4.2.2 Barres de translation

Les barres de translation sont des **Fl_Value_Slider**. Elles permettent de déterminer la translation selon les axes horizontaux et verticaux de la zone de visualisation. Ces trans-

lations sont déterminées comme étant un pourcentage de la longueur la plus longue des 3 directions principales des données. Le pas de translation est fixé à 1% de cette longueur.

4.2.3 Barres de rotation du volume

Les barres de rotation du volume sont composées d'une **FI_Roller** et une **FI_Output**. En effet, il n'existe pas de **FI_Value_Roller** et la valeur de l'angle de rotation est importante. Les **FI_Output** permettent d'afficher ces valeurs. Le pas de rotation est fixé à 1° , et le volume peut être tourné de -180° à $+180^\circ$.

4.2.4 Barre d'éclaircissement

La barre d'éclaircissement est une **FI_Value_Slider** et permet de fixer la valeur la plus haute des données à afficher. Autrement dit, les données comprises entre 0 et cette valeur sont affichées sur 256 niveaux de gris. Comme les données sont accumulées ce facteur doit être modifiable. En effet selon certains axes de rotation, très peu de données ou au contraire un très grand nombre peuvent être accumulées. Les valeurs choisies vont de 0 à 200000 par pas de 1.

BIBLIOGRAPHIE

- Andriole, K. P. (2004). Addressing the coming radiology crisis : The society for computer applications in radiology transforming the radiological interpretation process (trip) initiative. *Journal of Digital Imaging*, 17(4).
- Bajaj, C., Ihm, I., et Park, S. (2001). Visualization-specific compression of large volume data. Dans *Ninth Pacific Conference on Computer Graphics and Applications, 2001. Proceedings*, pages 212–222.
- Battle, G. (1987). A block-spin construction of ondelettes, part i : Lemarié functions. *Communications in Mathematical Physics*, 110 :601–615.
- Blinn, J. F. (1977). Models of light reflection for computer synthesized pictures. *Computer Graphics*, 11(2) :192–198.
- Cabral, B., Cam, N., et Foran, J. (1994). Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. Dans *VVS '94 : Symposium on Volume visualization. Proceedings*, pages 91–98. ACM Press.
- Cai, W. et Sakas, G. (2000). Drr volume rendering using splatting in shear-warp context. Dans *IEEE Nuclear Science Symposium Conference Record, 2000. Proceedings*, volume 3, pages 19/12–19/17 vol.3.
- Chen, W., Ren, L., Zwicker, M., et Pfister, H. (2004). Hardware-accelerated adaptive EWA volume splatting. Dans *IEEE Visualization 2004. Proceedings*.
- Daubechies, I. (1995). Where do wavelets come from? - a personal point of view. Rapport technique, Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University.
- Daubechies, I. et Sweldens, W. (1998). Factoring wavelet transforms into lifting steps. *J. Fourier Analysis and Applications*, 4(3) :245–267.
- Drebin, R. A., Carpenter, L., et Hanrahan, P. (1988). Volume rendering. Dans *SIGGRAPH '88 : 15th annual conference on Computer graphics and interactive techniques. Proceedings*, pages 65–74. ACM Press.
- Dudgeon, D. E. et Mersereau, R. M. (1990). *Multidimensional Digital Signal Processing*. Prentice Hall Professional Technical Reference.
- Gabor, D. (1946). Theory of communication. *Journal of the Institute of Electrical Engineers*, 93(3) :429–457.

- Gargantini, I., He, L., et Zhou, Y. (2000). Comparison of wavelets for volume rendering. Dans *Ninth International Conference on Geometry and Graphics. Proceedings*, pages 68–73, Johannesburg.
- Gelder, A. V. et Kim, K. (1996). Direct volume rendering with shading via three-dimensional textures. Dans *VVS '96 : Symposium on Volume visualization. Proceedings*, pages 23–30. ACM Press.
- Gross, M. H., Lippert, L., Dittrich, R., et Häring, S. (1995). Two methods for wavelet-based volume rendering. Rapport technique, Ecole polytechnique fédérale de Zurich.
- Gross, M. H., Lippert, L., Dreger, A., et Koch, R. (1994). A new method to approximate the volume rendering equation using wavelets and piecewise polynomials. Rapport technique, Ecole polytechnique fédérale de Zurich.
- Grosso, R., Ertl, T., et Joachim, A. (1996). Efficient data structures for volume rendering of wavelet-compressed data. Dans *Fourth International Conference in Central Europe on Computer Graphics and Visualization'96. Proceedings*, University of West Bohemia in Plzen.
- Guthe, S. et Strasser, W. (2004). Advanced techniques for high-quality multi-resolution volume rendering. *Computers & Graphics*, 28(1) :51–58.
- Guthe, S., Wand, M., Gonser, J., et Strasser, W. (2002). Interactive rendering of large volume data sets. Dans *VIS '02 : IEEE Visualization, 2002. Proceedings*, pages 53–60.
- Hadwiger, M., Berger, C., et Hauser, H. (2003). High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. Dans *IEEE Visualization 2003. Proceedings*, pages 301–308. IEEE.
- Horbelt, S., Unser, M., et Vetterli, M. (1999). Wavelet projections for volume rendering. Dans *Twentieth Annual Conference of the European Association for Computer Graphics (EUROGRAPHICS'99). Proceedings*, Milano, Italy.
- Ihm, I. et Lee, R. K. (1995). On enhancing the speed of splatting with indexing. Dans *IEEE Conference on Visualization, 1995. Visualization '95. Proceedings*, pages 69–76, 441.
- Kajiya, J. T. et Herzen, B. P. V. (1984). Ray tracing volume densities. Dans *SIGGRAPH '84 : 11th annual conference on Computer graphics and interactive techniques. Proceedings*, pages 165–174. ACM Press.
- Kilthau, S. et Möller, T. (2001). Splatting optimizations. Rapport technique, Simon Fraser University.

- Kim, T. et Shin, Y. (1999). An efficient wavelet-based compression method for volume rendering. Dans *Seventh Pacific Conference on Computer Graphics and Applications, 1999. Proceedings*, pages 147–156.
- Krishnan, K., Marcellin, M., Bilgin, A., et Nadar, M. (2004). Compression / decompression strategies for large volume medical imagery. Dans *SPIE Medical Imaging. Proceedings*, volume 5371.
- Kruger, J. et Westermann, R. (2003). Acceleration techniques for gpu-based volume rendering. Dans *IEEE Visualization, 2003. VIS 2003. Proceedings*, pages 287–292.
- Lacroute, P. et Levoy, M. (1994). Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics*, 28 :451–458.
- LaMar, E. C., Duchaineau, M. A., Hamann, B., et Joy, K. I. (2000). Multiresolution techniques for interactive texture-based volume visualization. Dans Erbacher, R. F., Chen, P. C., Roberts, J. C., et Wittenbrink, C. M., editors, *Visual Data Exploration and Analysis VII. Proceedings*, volume 3960, pages 365–374, Bellingham, Washington. SPIE, The International Society for Optical Engineering.
- Laur, D. et Hanrahan, P. (1991). Hierarchical splatting : a progressive refinement algorithm for volume rendering. Dans *Computer Graphics (SIGGRAPH'91). Proceedings*, volume 25, pages 285–288, Las Vegas.
- Lemarié, P. (1988). Ondelettes à localisation exponentielle. *Journal de Mathématiques Pures et Appliquées*, 67 :227–236.
- Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics Applications*, 8(3) :29–37.
- Levoy, M. (1990). Efficient ray tracing of volume data. *ACM Transactions On Graphics*, 9(3) :245–261.
- Lippert, L. (1998). *Wavelet-based volume rendering*. Ph.d. thesis, Swiss federal institute of technology.
- Lippert, L. et Gross, M. H. (1995). Fast wavelet based volume rendering by accumulation of transparent texture maps. *Computer Graphics Forum*, 14(3) :431–44.
- Mallat, S. G. (1989). A theory for multiresolution signal decomposition - the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :674–693.

- Meißner, M., Guthe, S., et Straßer, W. (2001). Higher quality volume rendering on pc graphics hardware. Rapport technique WSI-2001-12, Wilhelm Schickard Institute for Computer Science, Graphical-Interactive Systems (WSI/GRIS), University of Tübingen.
- Meißner, M., Huang, J., Bartz, D., Mueller, K., et Crawfis, R. (2000). A practical evaluation of popular volume rendering algorithms. Dans *VVS '00 : 2000 IEEE symposium on Volume visualization. Proceedings*, pages 81–90. ACM Press.
- Mora, B., Jessel, J.-P., et Caubet, R. (2002). A new object-order ray-casting algorithm. Dans *VIS '02 : IEEE Visualization, 2002. Proceedings*, pages 203–210.
- Mueller, K. et Yagel, R. (1996). Fast perspective volume rendering with splatting by utilizing a ray-driven approach. Dans *VIS '96 : 7th conference on Visualization '96. Proceedings*, pages 65–72, Los Alamitos, CA, USA. IEEE Computer Society Press.
- Neophytou, N. et Mueller, K. (2003). Post-convolved splatting. Dans *Symposium on Data visualisation 2003. Proceedings*, pages 223–230. Eurographics Association.
- Orchard, J. et Moller, T. (2001). Accelerated splatting using a 3d adjacency data structure. Dans *GI 2001 : Graphics interface 2001. Proceedings*, pages 191–200, Ottawa, Ontario, Canada. Canadian Information Processing Society.
- Phong, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM*, 18(6) :311–317.
- Piccand, S., Noumeir, R., et Paquette, E. (2005). Efficient visualization of volume data sets with region of interest and wavelets. Dans *SPIE Medical Imaging. Proceedings*.
- Piccand, S., Noumeir, R., et Paquette, E. (2006). Wavelet-based volume rendering with a region of interest. *Article en préparation*.
- Pinnamaneni, P. (2003). *Wavelet Based Volume Rendering*. M.sc. thesis, Faculty of Mississippi State University.
- Rezk-Salama, C., Engel, K., Bauer, M., Greiner, G., et Ertl, T. (2000). Interactive volume on standard pc graphics hardware using multi-textures and multi-stage rasterization. Dans *HWWS '00 : ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware. Proceedings*, pages 109–118. ACM Press.
- Rodler, F. (1999). Wavelet based 3d compression with fast random access for very large volume data. Dans *Seventh Pacific Conference on Computer Graphics and Applications, 1999. Proceedings*, pages 108–117.

- Said, A. et Pearlman, W. A. (1996). A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6 :243–250.
- Saladi, S., Pinnamaneni, P., et Meyer, J. (2003). Wavelets and textures with illumination for web-based volume rendering. Dans *High Performance Computing Symposium, 2003. Proceedings*.
- Sanchez, R. et Carvajal, M. (1998). Wavelet based adaptive interpolation for volume rendering. Dans *IEEE Symposium on Volume Visualization, 1998. Proceedings*, pages 127–134, 176.
- Schulze, J., Kraus, M., Lang, U., et Ertl, T. (2003). Integrating pre-integration into the shear-warp algorithm. Dans *2003 Eurographics/IEEE TVCG Workshop on Volume graphics. Proceedings*, pages 109–118. ACM Press.
- Shapiro, J. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12) :3445–3462.
- Swan, J.E., I., Mueller, K., Moller, T., Shareel, N., Crawfis, R., et Yagel, R. (1997). An anti-aliasing technique for splatting. Dans *IEEE Visualization '97. Proceedings*, pages 197–204, 544.
- Sweeney, J. et Mueller, K. (2002). Shear-warp deluxe : the shear-warp algorithm revisited. Dans *Symposium on Data Visualisation 2002. Proceedings*, pages 95–104. Eurographics Association.
- Sweldens, W. (1994). *Construction and Applications of Wavelets in Numerical Analysis*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium.
- Vetterli, M. et Herley, C. (1992). Wavelets and filter banks : Theory and design. *IEEE Transactions on signal processing*, 40(9) :2207–2232.
- Westenberg, M. A. et Roerdink, J. B. T. M. (2000). X-ray volume rendering by hierarchical wavelet splatting. Dans *15th International Conference on Pattern Recognition. Proceedings*, pages 163–166, Barcelona, Spain.
- Westermann, R. et Ertl, T. (1998). Efficiently using graphics hardware in volume rendering applications. Dans *SIGGRAPH '98 : 25th annual conference on Computer graphics and interactive techniques. Proceedings*, pages 169–177. ACM Press.
- Westermann, R. et Sevenich, B. (2001). Accelerated volume ray-casting using texture mapping. Dans *VIS '01 : IEEE Visualization, 2001. Proceedings*, pages 271–278.

Westover, L. (1990). Footprint evaluation for volume rendering. Dans *Computer Graphics SIGGRAPH Conference. Proceedings*, volume 24, pages 367–376.

Zwicker, M., Pfister, H., van Baar, J., et Gross, M. (2002). Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3) :223–238.