



Rapport de stage
Lien entre les librairies C++ ITK et ILR

KOMBAS Sofiane

Maîtrise informatique
Année universitaire : 2003-2004



Remerciements

Je tiens tout d'abord à remercier mon tuteur, David Sarrut, Maître de conférences, pour m'avoir accueilli dans son équipe et pour l'investissement très important qu'il a mis dans l'encadrement de mon stage.

Je voudrais aussi remercier l'ensemble du service de radiothérapie, et plus particulièrement Vlad Boldea, Jean-Noël Badel et Myriam Ayadi.

Table des matières

1	Introduction	3
1.1	Contexte du stage	3
1.2	La radiothérapie	3
1.3	Le Centre Léon Bérard	5
1.4	L'équipe de recherche	5
2	L'objectif du stage	5
2.1	Le déroulement du stage	5
2.2	Matériel	6
3	La conversion	6
3.1	ITK	6
3.1.1	Présentation d'ITK	6
3.1.2	Installation d'ITK	6
3.1.3	Détails d'ITK	7
3.2	Détails d'ILR	9
3.3	La conversion de ILR à ITK	10
3.4	La conversion de ITK à ILR	12
4	L'exemple "distance map"	14
5	Conclusion	15

1 Introduction

1.1 Contexte du stage

J'ai effectué dans le cadre de ma formation de maîtrise d'informatique, un stage de trois mois au sein de l'équipe du département de radiothérapie du centre Léon Bérard, centre de lutte contre le cancer de la région Rhône-Alpes. J'ai côtoyé des physiciens et dosimétristes du département ainsi des chercheurs du laboratoire LIRIS de l'université Lumière Lyon 2 délégués à l'unité recherche du département de radiothérapie du centre.

1.2 La radiothérapie

Cette technique de traitement des cancers utilise des radiations qui, lorsqu'elles traversent les tissus, provoquent des phénomènes d'ionisation. La profondeur atteinte par le rayonnement est liée à sa puissance. Les deux techniques de radiothérapie, röntgenthérapie et curiethérapie, ont des applications et des principes de fonctionnement différents. La première utilise des appareils qui produisent des rayons X : accélérateurs de particules (électrons, neutrons et protons) ; la seconde, les rayonnements gamma émis par des corps radioactifs implantés dans les cavités naturelles du corps (exothérapie) ou dans la tumeur elle-même (endothérapie). Les radiations ionisantes coupent directement ou indirectement l'ADN des noyaux cellulaires, bloquant la division des cellules, qui meurent, mais la radiothérapie s'attaque aux cellules normales aussi bien qu'aux cellules tumorales, c'est pourquoi il est nécessaire de définir avec précision le champ d'irradiation. Des progrès considérables ont été réalisés dans la détermination du contour et de l'emplacement de la tumeur à irradier grâce à l'amélioration des techniques d'irradiation (laser) et d'imagerie médicale (scanner). Les rayons présentent cependant toujours l'inconvénient d'endommager les tissus qu'ils traversent, provoquant brûlures et nécroses.

Le principal obstacle que les médecins peuvent affronter dans la réalisation du traitement est le positionnement précis et quotidien du patient sur la table de traitement.

L'objectif de la radiothérapie est de fournir une dose élevée de rayons X à la tumeur tout en maintenant une dose aux tissus sains environnants aussi basse que possible. Ces deux demandes contradictoires ont toujours été le principal défi pour les oncologistes, les physiciens et les radiothérapeutes depuis que le rayonnement a été employé dans la thérapie de cancer.

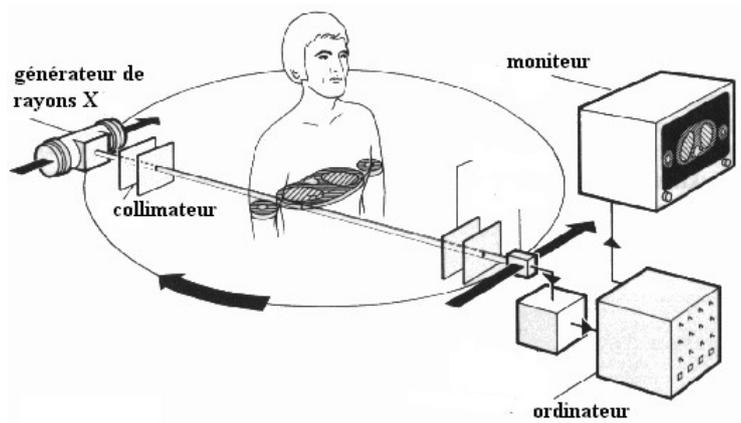


FIG. 1 – génération d'image médicale

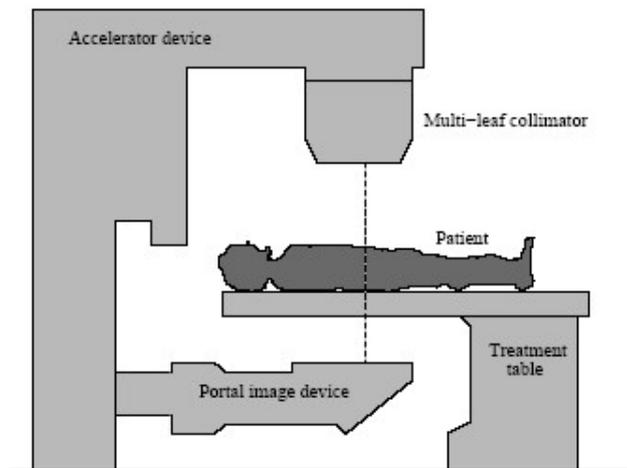


FIG. 2 – Electronique Portal Image Device

1.3 Le Centre Léon Bérard

Le centre de lutte contre le cancer Léon Bérard (CLB) est, comme vingt autres centres en France, un établissement de santé privé à but non lucratif, participant au service public hospitalier. A ce titre, ces centres sont financés par l'assurance maladie et contrôlés par le ministère de la Santé dans les mêmes conditions que les hôpitaux publics. Leur statut comporte explicitement une vocation de recherche appliquée et fondamentale.

1.4 L'équipe de recherche

L'équipe durant ces trois mois de stage était composée de :

- David Sarrut, mon tuteur de stage, maître de conférence et chercheur à plein temps en délégation auprès du centre Léon Bérard.
- Vlad Boldea, doctorant financé par une bourse Cifre (Elekta).
- Jean-Noël Badel doctorant physicien.
- Myriam Ayadi doctorant physicienne.
- Loïc Marteau, un autre stagiaire de maîtrise informatique.

2 L'objectif du stage

- ILR : Image Library for Radiotherapy, est une librairie de recalage et de segmentation d'image développée par David Sarrut.
- ITK : Insight ToolKit, ITK est un ensemble de bibliothèques de recalages et de segmentations d'images.

L'objectif de ce stage est de faire un lien entre les deux bibliothèques C++ ITK et ILR pour but d'appliquer les algorithmes de segmentations et de recalages de la bibliothèque ITK au format d'image ILR, et inversement.

2.1 Le déroulement du stage

Le déroulement de mon stage s'est fait de la manière suivante :

- Recherche bibliographique et étude de la bibliothèque ITK
- Installation de la bibliothèque ITK.
- Implémenter les deux méthodes de conversions.
- Application d'un algorithme d'ITK à des images au format ILR.

2.2 Matériel

Le développement de l'application de conversion c'est fait sous le système d'exploitation *GNU/Linux*, à l'aide du langage C++. En plus j'ai eu a disposition certains éléments qu'il a fallut rassemblé : les librairies ILR et ITK, Cmake et gengetopt.

3 La conversion

3.1 ITK

3.1.1 Présentation d'ITK

La librairie ITK (Insight ToolKit) est le produit d'un consortium (Insight Consortium) financé par le NIH (National Institute of Health) des Etats Unis, composé de trois industriels (Kitware, GE Corporate R&D, et Insightful) et de trois universités américaines (University of North Carolina, University of Utah et University of Pennsylvania). Il s'agit d'un produit Open-Source sous une licence de type Berkeley très peu contraignante pour ce qui est des produits qui s'appuient sur ces librairies. Il est ainsi possible de bâtir sa propre application autour d'un ou de plusieurs produits sous licence Berkeley sans avoir de contrainte de diffusion de ces propres sources.

ITK est développé à l'aide d'un environnement utilisant trois produits de la société Kitware également Open-Source sous licence de type Berkeley : CMake est un outil de configuration, Dart est un outil de testing et Cable et d'autres outils annexes permettent le wrapping des classes C++ vers d'autres langages, les informations sur ces sujets peuvent être trouvées dans [1] [2] [3]. ITK est un ensemble de librairies de recalage et de segmentations d'images mais ne prend pas en charge les aspects visualisation et d'interface graphiques. Ces fonctionnalités sont laissées à d'autres produits du marché, par exemple QT ou FLTK pour l'aspect interface graphique et VTK pour l'aspect visualisation. Cependant, ITK est suffisamment ouvert pour s'interfacer avec tout autre produit.

3.1.2 Installation d'ITK

L'installation d'ITK exige au moins la version 1.6 de CMake.L'utilisation de ce dernier exige que nous lui fournissant deux informations : là ou le code source est localisé (ITK_SOURCE_DIR), et ou le code binaire doit être produit (ITK_BINARY_DIR).

```
mkdir Insight-binary
cd Insight-binary
ccmake ../Insight
ITK est multiplates-formes (Unix, Windows et MacOSX).
```

3.1.3 Détails d'ITK

L'architecture de code d'ITK est fortement basée sur la programmation générique, c'est à dire l'utilisation des templates C++.

La classe de base qui contrôle la lecture et l'écriture des images est itk : :ImageIOBase. Cette classe a été dérivée afin de lire :

- PNG files avec itk : :PNGImageIO.
- JPEG files avec itk : :JPEGImageIO.
- TIFF files avec itk : :TIFFImageIO.
- DICOM files avec itk : :DICOMImageIO2.
- Analyze files avec itk : :AnalyzeImageIO.
- MetaImage files avec MetaImageIO.
- GIPL files avec itk : :GiplImageIO.
- Raw files avec itk : :RawImageIO.
- Stimulate files avec itk : :StimulateImageIO.
- VTK files avec itk : :VTKImageIO.
- GE 4 files avec itk : :GE4ImageIO.
- GE 5 files avec itk : :GE5ImageIO.
- GE Adw files avec itk : :GEAdwImageIO.
- SiemensVision files avec itk : :GEAdwImageIO.

ImageIOBase est une classe qui permet de lire et ou d'écrire des données d'image d'un format particulier (tels que PNG ou RAW).ImageIOBase encapsule la lecture et l'écriture des données.ImageIOBase est utilisé par la class ImageFileReader (lecture de données) et par la class ImageFileWriter (écriture de données).Les classes ImageSeriesReader et ImageSeriesWriter sont utilisées pour lire et écrire des données quand celles-ci sont représentées par une série de fichiers.

La classe Image suit l'esprit de la programmation générique, ou les types sont séparés du comportement algorithmique de la classe.

The Insight toolkit n'impose aucun format particulier d'image. Le format de fichier choisi pour convertir le format VOX est le format MetaImage.

La classe itk::Image suit l'esprit de la programmation générique, ou les types sont séparés du comportement algorithmique de la classe. ITK accepte des images avec n'importe quel type de pixel et n'importe quelle dimension spatiale. Dans ITK, les images existent en combinaison avec une ou plu-

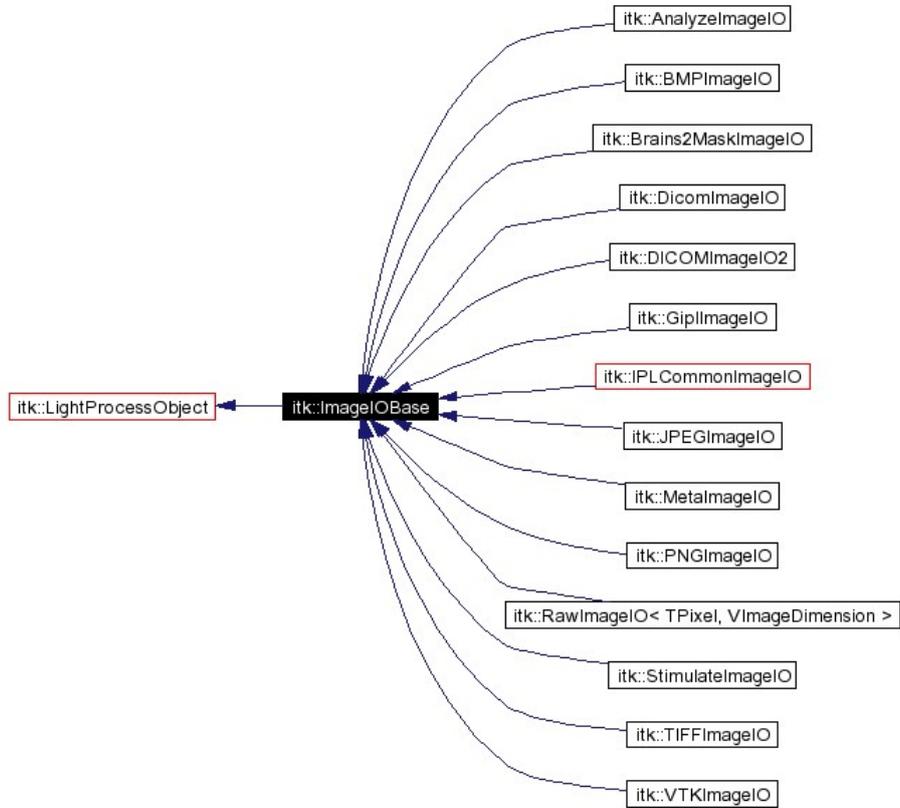


FIG. 3 – Diagramme de collaboration de la classe ImageIOBase

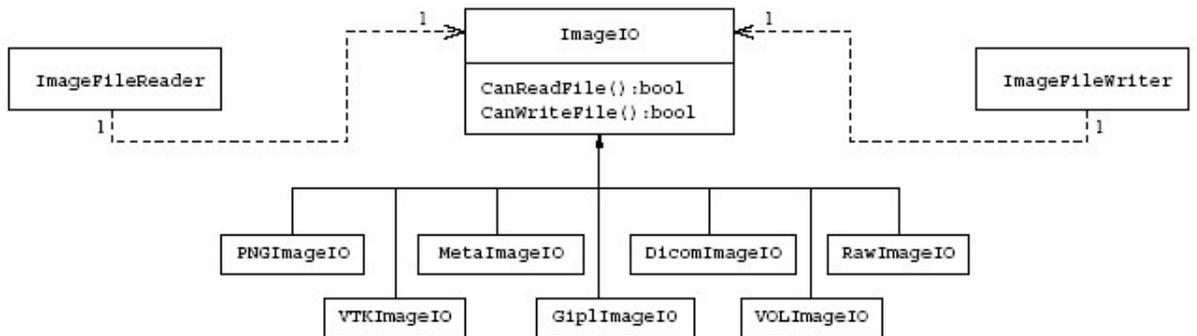


FIG. 4 – Diagramme de collaboration de la classe ImageIO

sieurs régions. Une région est un sous-ensemble de l'image et indique une partie de l'image qui peut être traitée par d'autres classes dans le système. Une des régions les plus communes est la LargestPossibleRegion, qui définit l'image en sa totalité. D'autres régions importantes trouvées dans ITK sont le BufferedRegion, qui est la partie de l'image réellement maintenue dans la mémoire, et le RequestedRegion, qui est la région demandée par un filtre ou toute autre classe en fonctionnant sur l'image.

Une région est définie par deux classes : itk : :Index et itk : :Size. L'origine de la région dans l'image à laquelle elle est associée est définie par Index. La taille de la région est définie par Size. L'index indique le pixel initial de l'image.

Quand une image est créée manuellement, l'utilisateur est responsable de définir la taille d'image et l'index auxquels la grille d'image commence. Ces deux paramètres permettent de traiter des régions choisies.

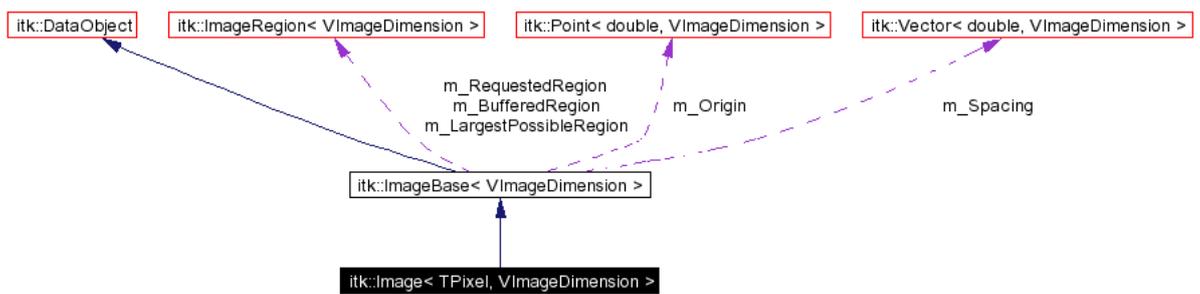


FIG. 5 – Diagramme de collaboration de la classe Image

3.2 Détails d'ILR

Comme la librairie ITK, ILR est fortement basée sur la programmation générique. Ces principales classes sont GridBase, Grid. Dans la classe Grid, est définie un certain nombre de fonctions qui permettent de créer des images ILR ainsi à accéder aux données et de les modifier.

La classe GridReaderWriter permet de lire et d'écrire les fichiers VOX

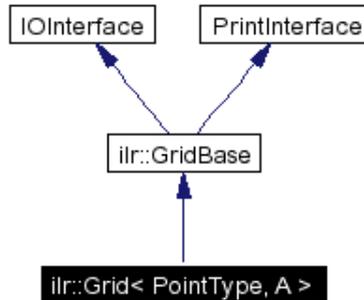


FIG. 6 – Diagramme de collaboration de la classe Grid

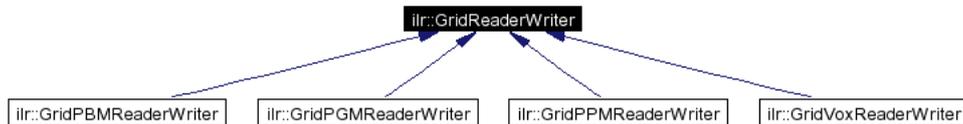


FIG. 7 – Diagramme de collaboration de la classe GridReaderWriter

3.3 La conversion de ILR à ITK

Pour localiser les tumeurs par rapport aux organes, il est nécessaire d’obtenir une image volumique du patient. Les images traitées au cours du développement de l’application est de dimension trois.

Le format de fichier VOX, présente au début des informations sur l’image ; sa dimension, sa taille et la taille des voxels, dont voici un exemple d’en-tête :

```

VOX
## generated on 13/3/2003
128 128 55
3.75 3.75 5
  
```

La parti qui suit, présente les données de l’image, les informations portées par les voxels indiquent des niveaux de densité de matière exprimée en unités Hounsfield, liés à la densité réelle. Ces intensités peuvent ensuite être converties en niveau de gris pour l’affichage des images.

Le format de fichier MetaImage, c’est un format assez simple mais puissant se composant d’en-tête et d’une section binaire de données, dont voici un exemple d’en-tête :

```

NDims = 3
  
```

```
DimSize = 181 217 181
ElementType = MET_UCHAR
ElementSpacing = 1.0 1.0 1.0
ElementByteOrderMSB = False
ElementDataFile = brain.raw
```

- NDims indique la dimension de l'image. ITK peut manipuler des images de dimension arbitraire.
- ElementType indique la taille de l'image en pixels le long de chaque direction.
- ElementType indique le type de pixel
- ElementSpacing indique la séparation physique entre le centre d'un pixel et le centre du prochain pixel le long de chaque direction dans l'espace. Les unités utilisées sont des millimètres.
- ElementByteOrderMSB indique si les données sont encodées dans l'ordre little ou big endian.
- ElementDataFile est le nom du fichier contenant les données binaires de l'image. Ce fichier doit être dans le même répertoire que celui contenant l'en-tête.

Les principes fonctions :

```
template<typename PointType>
void ilr2itk (ilr::Grid<PointType> * input,typename itk::Image<PointType,
3>::Pointer output, const bool copy)
```

Cette fonction permet de convertir un `ilr : :Grid<PointType>` en `itk : :Image<PointType, 3>` le booléen `copy` dans cette fonction permet de choisir l'une des deux méthodes :

- Une copie de données, on réserve une zone de mémoire de taille égale au nombre de voxels contenu dans l'image ILR et ensuite on copie les données, et enfin on assigne le pointeur de donnée à cette zone de mémoire.
- Le partage du pointeur de données, on récupère le pointeur de données de l'image ILR qui pointe au début de la zone de mémoire, qui devient le pointer de données de l'image ITK.

```
template<typename PointType>
void newilr2itk( GridBase* in, itk::ImageBase<3>::Pointer ItkOut, bool
copy, MetaImage& metaIm)
```

Cette fonction permet de créer un `MetaImage`, en utilisant le `Write` qui va permettre d'écrire l'image dans le disque dur, dans les deux fichiers RAW et

mdh.

Pour les exécutable, l'outil Gengetopt[9] a été utilisé pour la gestion de la ligne de commande :

Il crée une fonction C qui vérifie automatiquement la ligne de commande et l'analyse pour d'autres programmes. Les résultats sont retournés dans une "struct" (structure C).

C'est un outil très utile pour les programmeurs C/C++, puisqu'il se charge du travail fastidieux mais important de l'analyse de la ligne de commande. Le programmeur spécifie simplement les options désirées, si elles sont optionnelles ou obligatoires, et s'il leur faut un paramètre. Le code C généré par gengetopt est ajouté au programme et un simple appel à la fonction valide et évalue la ligne de commande.

Exemples d'exécutable :

```
ilr2itk -i toto.vox -o titi.mhd
ilr2itk -c -v -i toto.vox -o titi.mhd
```

Les options :

-i : le fichier vox en entrée (obligatoire).

-o : le fichier mhd en sortie (obligatoire).

-c : on copie les données. Par défaut c'est une affectation de pointeur de données.

-v : affiche les informations de l'image d'entrée et de sortie.

3.4 La conversion de ITK à ILR

Une des difficultés rencontrées au cours de l'implémentation de cette conversion, est que la fonction ImageFileReader exige de déclarer le type de l'image c'est à dire de déclarer le type de pixel ainsi la dimension de l'image avant de la lire.

```
typedef itk::Image< PixelType, Dimension > ImageType ;
typedef itk::ImageFileReader< ImageType > ReaderType ;
ReaderType::Pointer reader = ReaderType::New() ;
reader->SetFileName( filename ) ;
```

Pour résoudre ce problème, le format MetaImage a été utilisé qui contient un lecteur générique.

Les principales fonctions :

```
template<typename PointType>
void itk2ilr( typename itk::Image<PointType,3>::Pointer input,
ilr::Grid<PointType>* IlrImage)
```

Cette fonction permet de convertir un itk::Image en ilr::Grid.

```
template<typename PointType>
void metaImage2image( MetaImage& in, itk::Image<PointType,3>& out)
Cette fonction permet de convertir un itk : :MetaImage en itk : :Image.
```

Exemples d'exécutables :

```
itk2ilr -i toto.vox -o titi.mhd
itk2ilr -v -i toto.vox -o titi.mhd
```

L'application ViewVox permet la visualisation des images VOX.Des testes ont été effectués pour vérifier la conversion.

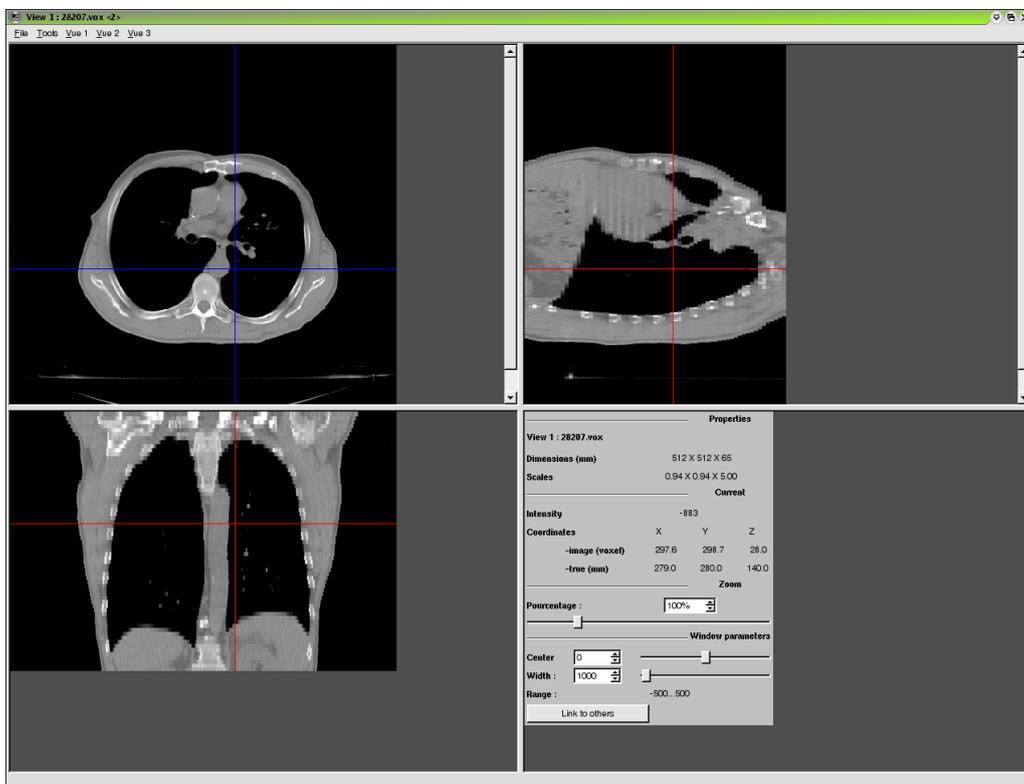


FIG. 8 – ViewVox

4 L'exemple "distance map"

L'exemple des "distance map", est un algorithme de Danielsson implémenté par ITK et qui consiste à calculer la distance euclidienne minimale, exemple :

Soit une image binaire (4x4)

0	0	0	0
0	1	0	0
0	0	0	0
0	0	1	0

distance map

$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
1	0	1	2
$\sqrt{2}$	1	1	$\sqrt{2}$
2	1	0	1

Pour appliquer cet algorithme au format d'image ILR, on a procédé de la manière suivante :

1. lire le fichier VOX.
2. Convertir en ITK.
3. Appliquer l'algorithme de Danielsson.
4. Ecrire le resultat en ILR.

5 Conclusion

Ce qui a été fait : la conversion du format VOX au format RAW avec la possibilité de choisir la manière de copie de données, ainsi la conversion du format RAW au format VOX. Une application de l'algorithme " distance-map ", qui est un algorithme implémenté par la librairie ITK, sur le format ILR.

Ce qui peut être amélioré : Dans la conversion de ITK vers ILR, une seule méthode de copie de données a été implémenté, la méthode qui consiste à partager le pointeur de données peut être ajouter, ce qui permettra un gain de mémoire et une exécution plus rapide de l'application.

Ce qui n'a pas été fait : La visualisation du format RAW, nécessite l'étude de librairie C++ VTK (The Visualization ToolKit) des informations sur cette librairie peuvent être trouvées dans [4] .

Références

- [1] <http://www.cmake.org>
- [2] <http://public.kitware.com/Dart/HTML/Index.shtml>
- [3] <http://public.kitware.com/Cable/HTML/Index.html>
- [4] <http://www.vtk.org/>
- [5] <http://www.itk.org>
- [6] <http://www.cs.utah.edu/~cates/InsightTalk/>
- [7] <http://caddlab.rad.unc.edu/publications/Tutorials/Summer-02/index.html>
- [8] http://www.vialab.org/methods_course/
- [9] <http://www.gnu.org/software/gengetopt/>