

Compte rendu du TP n° 1 d'imagerie 3D

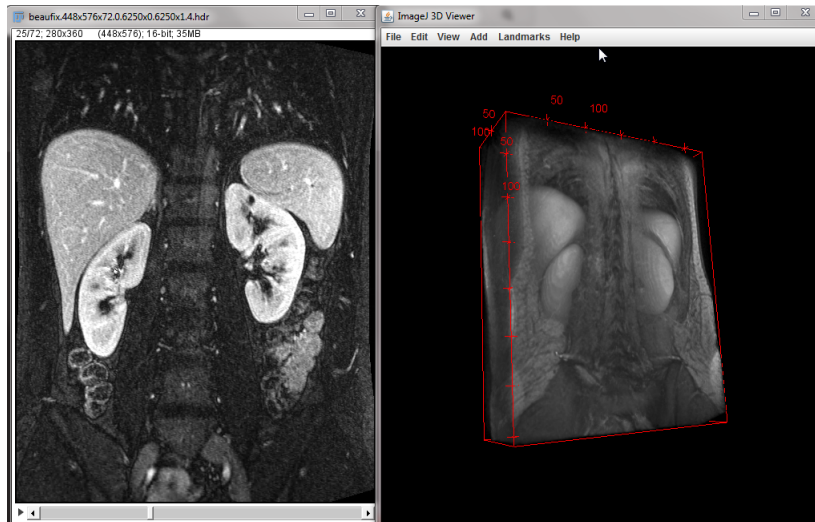
Imagerie médicale 3D

Thibaut Castanié
Master IMAGINA

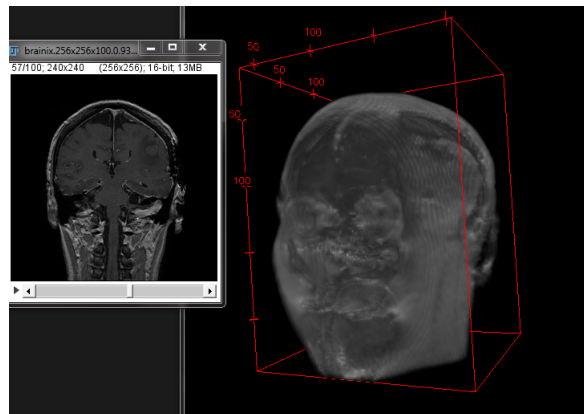
12 mars 2015

1 Découverte des images proposées

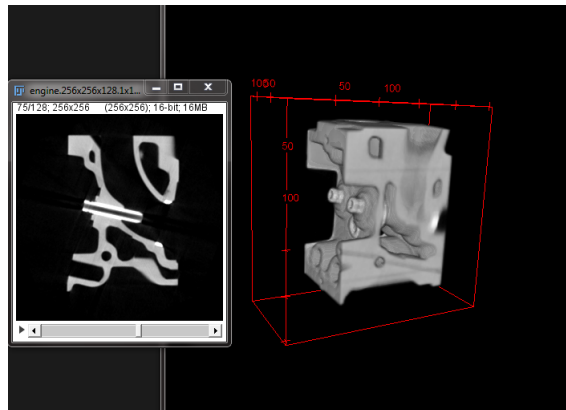
Dans un premier temps, les images proposées ont été ouvertes avec Fiji afin de visualiser les différentes couches d'images. Un plugin de rendu 3D permet de les superposer avec un effet de transparence, afin d'obtenir une représentation 3D de l'objet.



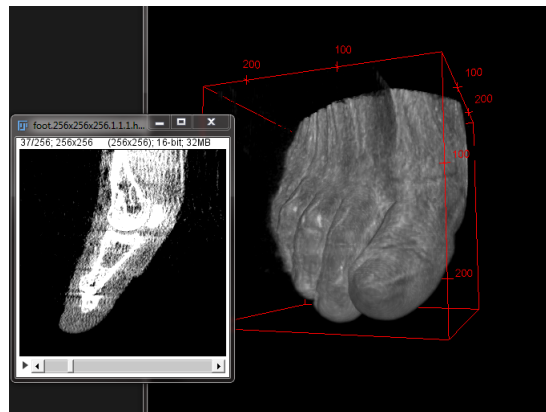
beaufix.img représentant l'intérieur du ventre humain



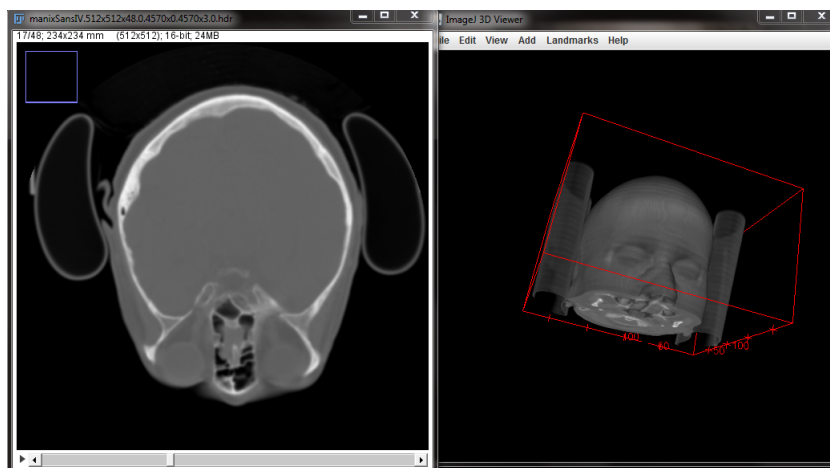
brainix.img représentant la tête d'un homme



engine.img représentant un moteur de voiture



foot.img représentant un pied humain



manix.img représentant le cerveau d'un homme

2 Programme de lecture

Afin de pouvoir manipuler les images, il a fallu créer un programme permettant de lire directement le fichier contenant les images. Celui-ci est composé d'une suite de valeur codant le niveau de gris du voxel concerné. Cette valeur est composée de deux **unsigned short** de deux bits chacun. Les valeurs sont classées de façon à lire une image après l'autre en fonction de l'axe Z de profondeur.

Faut-il écrire E6 ou 6E ? La réponse à cette question varie en fonction du logiciel utilisé lors de la capture des images ou du rendu de celles-ci. Dans notre cas, il a fallu inverser la position des **short** afin d'avoir des valeurs cohérentes.

Après implémentation de l'algorithme de lecture, une série de tests a été effectuée pour vérifier la justesse du programme. Les données suivantes ont été obtenues :

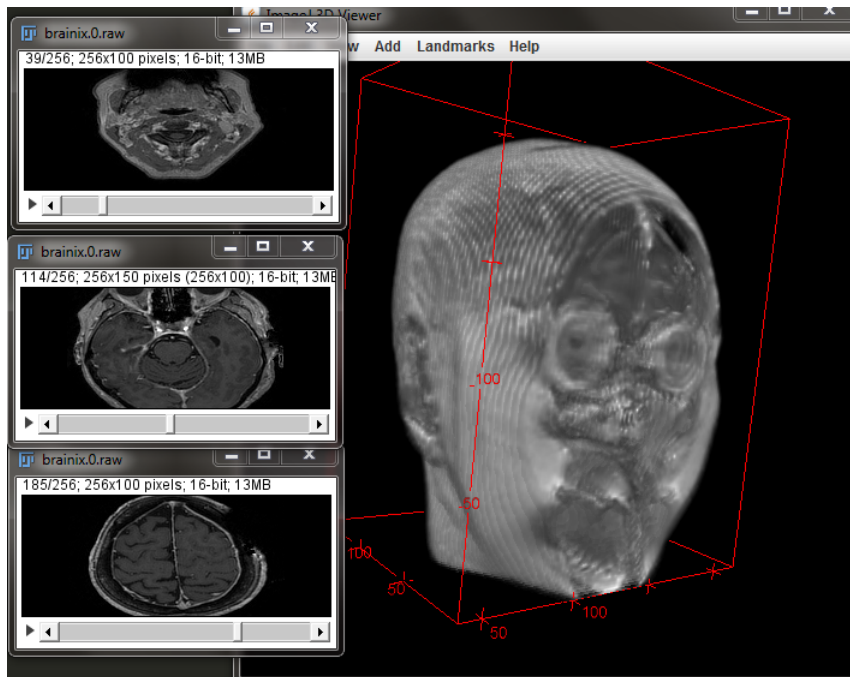
- beaufix : max=334, I(200,200,20)=14
- branix : max=1715, I(200,200,20)=4
- engine : max=255, I(200,200,20)=3
- foot : max=255, I(200,200,20)=0
- manix : max=2979, I(200,200,20)=1029

Code en annexe

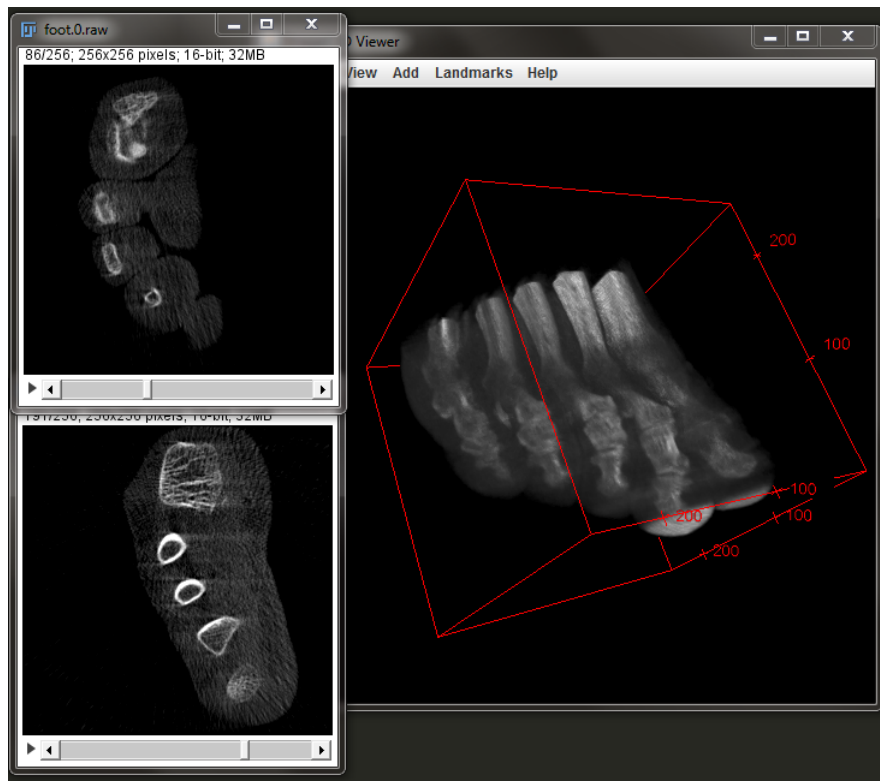
3 Programme de Volume Rendering

Afin d'utiliser le programme de lecture créé, un code permettant de recréer les données en entrée, mais cette fois en prenant l'axe Z de hauteur en tant qu'axe de profondeur.

Il est possible d'effectuer diverses actions sur ces données. Par exemple, le calcul de la moyenne des couches permet d'obtenir un visuel en deux dimensions qui donne une bonne idée de l'objet. En utilisant le module de rendu 3D de Fiji, on peut aussi obtenir une bonne représentation en trois dimensions de l'objet, navigable en temps réel.



Visualisation de brainix, après redimensionnement du voxel en Z à 1,5



Visualisation de foot

4 Qu'est ce que whatisit ?

En effectuant la moyenne des valeurs des pixels, sur chaque couche, en fonction de l'axe Z de profondeur, on obtient une image représentant une petite écrevisse de rivière.



Image obtenue après moyenne sur l'axe Z

Annexe - Code

```
#include <iostream>
#include <stdio.h>
#include <cstdlib>
#include <math.h>
#include <string>
using namespace std;

int tailleX = 512;
int tailleY = 512;
int tailleZ = 48;
unsigned short * buffer;

int getValue(unsigned short * buffer,int x, int y, int z){
    int num = z*(tailleX*tailleY)+(y*tailleY)+x;
    int oct1 = buffer[num]%256;
    int oct2 = buffer[num]/256;
    int valPixel = oct1*256+oct2;
    return valPixel;
}

int main(int argc, char const *argv[]){
    long taille;
    size_t result;
    FILE * imgFile;
    FILE * stockFile;
    short valeurPixel = 0;
    short max = 0;
    short octet1;
    short octet2;
    //imgFile = fopen (".\\BEAUFIX/beaufix.448x576x72.0.6250x0.6250x1.4.img","rb");
    //imgFile = fopen (".\\BRAINIX/brainix.256x256x100.0.9375x0.9375x1.5.img","rb");
    //imgFile = fopen (".\\engine/engine.256x256x128.1x1x1.img","rb");
    //imgFile = fopen (".\\FOOT/foot.256x256x256.1.1.1.1.img","rb");
    imgFile = fopen (".\\MANIX/manixSansIV.512x512x48.0.4570x0.4570x3.0.img","rb");
    //imgFile = fopen (".\\WHATISIT/whatisit.301x324x56.1.1.1.4.img","rb");
    if (imgFile!=NULL){
        fseek (imgFile , 0 , SEEK_END);
        taille = ftell (imgFile);
        rewind (imgFile);
        unsigned short *buffer = new unsigned short[taille];
        result = fread (buffer,2,taille,imgFile);
        cout << result << endl;
```

```

    if (result == taille/2) {
        for (int i = 0; i < result; i++){
            octet1 = buffer[i]%256;
            octet2 = buffer[i]/256;
            int valeurPixel = octet1*256+octet2;
            if (valeurPixel > max){
                max = valeurPixel;
            }
        }
        cout << "Max :\t" << max << endl; // valeur maximum
    }
    int retour = getValue(buffer,200,200,20);
    cout << retour << endl;

    stockFile = fopen ("manix.0.raw","wb");
    unsigned short *render = new unsigned short[tailleX*tailleZ*tailleY];
    int cpt=0;
    for (int k = 0; k < tailleY; k++){
        for(int i = 0; i<tailleZ;i++){
            for (int j = 0; j < tailleX; j++){
                render[cpt] = getValue(buffer,j,k,i);
                cpt++;
            }
        }
    }
    fwrite(render,2,tailleX*tailleZ*tailleY,stockFile);

    fclose (stockFile);
    fclose (imgFile);
    free (buffer);
}
return 0;
}

```