

## TP Imagerie 3D n° 1 (3 heures)

Gérard Subsol, 5 mai 2017

- Les programmes doivent être écrits en **C/C++** en n'utilisant que des **fonctions classiques** (stdio, stdlib, math).
- **Le TP est noté : le compte-rendu doit être envoyé sous forme électronique à : [gerard.subsol@lirmm.fr](mailto:gerard.subsol@lirmm.fr).**
  - Il faut rendre une première version du **compte-rendu avant 13h00** (80 à 100% de la note).
  - Si vous le souhaitez, une seconde version (0 à 20% de la note) pourra être envoyée avant **le jeudi 12 mai 2017 (minuit)**
- Le compte-rendu doit inclure **votre nom**, quelques lignes d'explication sur l'algorithme, des captures d'écran (ici, la visualisation en Volume Rendering des différentes images 3D, en particulier whitisit) et le code source intégral (avec quelques commentaires). Le tout doit être sous la forme d'un **unique fichier pdf**.
- Le TP est individuel.
- La participation active pendant le TP pourra aussi être prise en compte.
- **Tout plagiat sera lourdement sanctionné.**

1. Découvrir les images 3D orange, INCISIX et t1-head (format Analyze) à l'aide de Fiji.

2. **Ecrire en C/C++ un programme** avec les fonctions suivantes :

- Lecture d'une image img qui est dans un format brut (sans en-tête), codée en unsigned short (2 octets)
- Stockage en mémoire de l'image
- getValue(i,j,k) qui renvoie la valeur du voxel (i,j,k)
- Affichage de la valeur minimale et maximale des voxels de l'image
- Affichage de l'intensité du voxel milieu de l'image.

On pourra utiliser les paramètres suivants :

<imageFile> <dimX> <dimY> <dimZ>

3. Tester sur les images (dans Fiji : ctrl+H en incluant toutes les images vous donne le min/max) :

- orange min=0 max=228 I(128,128,32)=15
- INCISIX min=0 max=4095 I(256,256,83)=274
- T1-head min=0 max=885 I(128,128,64)=80

4. **Ecrire en C/C++ un programme de Volume Rendering** (MIP, AIP, MinIP) suivant la direction axiale z. Le résultat sera une image qui pourra être au format brut (sans en-tête) et codée en unsigned short (2 octets). Elle pourra être lue par Fiji.

On pourra utiliser les paramètres suivants :

<imageFile> <dimX> <dimY> <dimZ> <resultFile> <visuFlag : 1=mip, 2=aip, 3=minip>

5. Tester sur les images : orange / INCISIX / T1-head

6. **Défi** : qu'est-ce que whitisit ?

7. On peut étendre le programme en proposant les directions x et y.

**A. Données disponibles à :** <http://www.lirmm.fr/~subsol/GMIN215/>

**B. Fiji :** <http://fiji.sc/>

Pour lire une image 3D au format Analyze : Fille/Open ou File/Import/Analyze puis le fichier .hdr

Pour lire une image 2D brute : File/Import/Raw puis rentrer les paramètres (16 bits par exemple).

**C. Stockage de l'image**

On supposera que l'image est lue coupe par coupe et qu'elle est balayée par ligne suivant :

```
y\x ----->
| 7 8 9
| 4 5 6
v 1 2 3
```

**D. Rappels de fonction C de lecture/écriture de données binaires**

**FILE \*fopen(const char \*filename, const char \*mode);**

Opens the filename pointed to by filename. The mode argument may be one of the following constant strings:

**rb** read binary mode

**wb** write binary mode (truncates file to zero length or creates new file)

**size\_t fread(void \*ptr, size\_t size, size\_t nmemb, FILE \*stream);**

Reads data from the given stream into the array pointed to by *ptr*. It reads *nmemb* number of elements of size *size*. The total number of bytes read is (**size\*nmemb**).

On success, the number of elements read is returned. On error or end-of-file the total number of elements successfully read (which may be zero) is returned.

**size\_t fwrite(const void \*ptr, size\_t size, size\_t nmemb, FILE \*stream);**

Writes data from the array pointed to by *ptr* to the given stream. It writes *nmemb* number of elements of size *size*. The total number of bytes written is (**size\*nmemb**).

On success, the number of elements written is returned. On error the total number of elements successfully written (which may be zero) is returned.