TP Imagerie 3D nº 1 (3 heures)

Gérard Subsol, 6 mars 2015

- Les programmes doivent être écrits en **C/C++** en n'utilisant que des **fonctions classiques** (stdio, stdlib, math).
- **Le TP est noté**. Il faut rendre un **compte-rendu avant** le prochain cours prévu **le vendredi 13 avril 2015** sous **forme électronique à : gerard.subsol@lirmm.fr**
- Le compte-rendu doit inclure quelques lignes d'explication sur l'algorithme et le code, des captures d'écran (ici, la visualisation en Volume Rendering des différentes images 3D, en particulier whatisit) et le code source intégral. 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 : BEAUFIX, BRAINIX, engine, FOOT, MANIX à l'aide de MicroView (voir D.).

2. **Ecrire en C/C++ un programme** avec les fonctions suivantes :
   - Lecture d'images img qui sont dans un format brut (sans en-tête) et codées en unsigned short (2 octets) (voir B. et C.).
   - 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.

   On pourra utiliser les paramètres suivants :
   <imageFile> <dimX> <dimY> <dimZ>

3. Tester sur les images (les intensités sont lues dans MicroView) :
   - BEAUFIX : min=0 max=334 I(200,200,200)=14
   - BRAINIX : min=0 max=1715 I(200,200,20)=4
   - engine : min=0 max=255 I(200,200,20)=3
   - FOOT : min=0 max=255 I(200,200,20)=0
   - MANIX : min=0 max=2979 I(200,200,20)=1029

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 MicroView (voir A.).
   On pourra utiliser les paramètres suivants :
   <imageFile> <dimX> <dimY> <dimZ> <resultFile> <visuFlag : 1=mip, 2=aip, 3=minip>

   Attention, pour lire l'image brute avec MicroView, il faut la sauvegarder sous le nom <xxx.**0.raw**>

5. Tester sur les images : engine / FOOT / BEAUFIX / BRAINIX / MANIX / orange

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

A. MicroView *http://microview.sourceforge.net/* (installé sur les machines de TP)

- Pour lire une image 3D : File/Open/Interfile format (*.hdr) lit directement les fichiers hdr (en-tête) + img (données brutes)

- Pour lire une image 2D brute : File/Image import/
  - Sélectionner le fichier raw
  - Cliquer sur Raw Image
  - Rentrer X size, Y size,

B. Stockage de l'image

On supposera que l'image est lue coupe par coupe et qu'elle est balayée par lignes.

C. 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.

D. Données disponibles à : *http://www.lirmm.fr/~subsol/GMIN215/*