

## TP Imagerie 3D n° 2 (3 heures)

- Le TP est **individuel**.
- 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) avant 13h00.**
- Le compte-rendu doit inclure **votre nom**, quelques lignes d'explication sur l'algorithme, des captures d'écran (ici, le maillage, si possible superposé sur une vue MPR) et le code source intégral. Le tout doit être sous la forme d'un **unique fichier pdf**.
- La participation active pendant le TP pourra aussi être prise en compte.
- **Tout plagiat sera lourdement sanctionné.**

### 1. Reprendre le programme C/C++ du TP précédent avec les fonctions suivantes :

- *Lecture des images img au format brut (sans en-tête) et codées en unsigned short (2 octets).*
- *Stockage en mémoire de l'image*
- *Fonction `getValue(i,j,k)` qui renvoie la valeur du voxel (i,j,k)*

### 2. Calcul d'une iso-surface (de seuil s) d'une image 3D sous la forme d'un maillage 3D

Pour cela, on programmera une version simplifiée de l'algorithme du Marching Cubes vu en cours :

- Parcours de tous les voxels de l'image (en ne prenant pas en compte les voxels du bord).
- Pour chaque voxel, calcul des coordonnées réelles (c'est-à-dire en tenant compte de la taille du voxel) des 8 sommets de la façon suivante :

```

/* (i,j,k) is the center of the voxel */
/* Real coordinates of the center: (i*sizeX, j*sizeY, k*sizeZ)
/* The X vertices abscissae of the voxel corners are then
   (i-0.5)*sizeX and (i+0.5)*sizeX */

/*      z^      7 --- 6          */
/*      I      /I      /I          */
/*      I      4 --- 5 I          */
/*      I      I 3 --I 2          */
/*      I      I/      I/          */
/*      I/      0 --- 1          */
/*      +-----> x          */

/* 8 vertices          */

```

- Si la valeur du voxel (i,j,k) est supérieure au seuil s :
  - Alors si la valeur d'un des voxels voisins (on raisonnera en 6-voisinage) est inférieure au seuil, ajouter au fichier résultat les 2 triangles qui forment la face entre les deux. Cette face sera une approximation de l'isosurface. Les triangles seront créés au format STL (voir annexe).

Justifier pourquoi on ne crée pas deux fois les mêmes triangles entre deux voxels.

**Attention !** Afin d'avoir un affichage des faces cohérent, les points de la face doivent toujours être dans un ordre tel que quand on calcule la normale au triangle par l'orientation canonique (on visse), celle-ci doit être « sortante » (c'est-à-dire aller du centre du voxel vers l'extérieur). Par exemple, la face éventuellement créée entre le voxel et le voxel du dessus devra être formée des triangles [456] et [674] dans le schéma ci-dessus.

On pourra utiliser les paramètres suivants :

```
<imageFile> <dimX (int)> <dimY (int)> <dimZ (int)> <sizeXVoxel (float)> <sizeYVoxel (float)>  
<sizeZVoxel (float)> <resultFile (string)> <threshold (unsigned short)>
```

### 3. Tester sur les images 3D, en particulier :

engine :    seuil=200 (petit fichier)  
             seuil=100 (gros fichier)

statueLeg : seuil=50 (petit fichier)

MANIX :    seuil=900 (très gros fichier)  
             seuil=1100 (très gros fichier)

### 4. Trouver empiriquement le seuil « optimal » pour whatisit et extraire l'isosurface.

On pourra visualiser les maillages 3D avec Fiji ou MeshLab (voir annexe).

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

- Plugins/3D Viewer permet de visualiser l'image en Volume (rendering) ou Orthoslice (mettre 1 à Resampling factor pour ne pas sous-échantillonner l'image). Attention, il faut d'abord convertir l'image en 8 bits par Image/Type/8-bit.
- Image/Adjust/Threshold permet de visualiser les effets d'un seuillage.
- On peut alors lire un maillage surfacique au format stl par : File/Import Surfaces/STL

**B. MeshLab** <http://meshlab.sourceforge.net/> est un programme très puissant de traitement et de modélisation de maillages 3D.**C. Format STL**

STL (Standard Tessellation Language) is a file format (binary or ASCII) native to the stereolithography CAD software created by 3D Systems. This file format is supported by many other software packages; it is widely used for rapid prototyping and computer-aided manufacturing.

An ASCII STL file begins with the line:

*solid name*

where name is an optional string.

The file continues with any number of triangles, each represented as follows:

```
facet normal 0 0 0      (the normal coordinates are optional)
outer loop
vertex v1x v1y v1z
vertex v2x v2y v2z
vertex v3x v3y v3z
endloop
endfacet
```

where each n or v is a floating point number.

The file concludes with:

endsolid name

(Source : [http://en.wikipedia.org/wiki/STL\\_file\\_format](http://en.wikipedia.org/wiki/STL_file_format) )

**D. Données disponibles à :** <http://www.lirmm.fr/~subsol/HMIN211/>