

TP2 Imagerie 3D

Yasmine Khodja

1. Calcul d'iso-surface:

En appliquant l'algorithme donné sur un voisinage de 6 faces:

L'algorithme consiste à vérifier si l'intensité du voxel est supérieure au seuil, si c'est le cas on vérifie si l'intensité de ses voisins est inférieure au seuil, si c'est le cas on dessine les triangles que forme la face qu'ils ont en commun.

```
void seuillage(unsigned short* image, unsigned short seuil, int dimX, int dimY, int dimZ, float
sizeX, float sizeY, float sizeZ, char nomFich[]){
```

```
FILE* fichier;
```

```
unsigned short voxel = 0;
```

```
fichier = fopen(nomFich,"w");
```

```
fprintf(fichier,"solid name\n");
```

```
for(int i=1; i<dimZ-1; i++){
```

```
for(int j=1; j<dimX-1; j++){
```

```
for(int k=1; k<dimY-1; k++){
```

```
//printf("(%d,%d,%d) = %hu\n",i,j,k, getValue(image,i,j,k,dimX,dimY,dimZ));
```

```
if(getValue(image, i, j, k, dimX, dimY, dimZ) >= seuil){
```

```
//face à gauche
```

```
if(getValue(image, i, j-1, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
}
```

```
//face à droite
```

```
if(getValue(image, i, j+1, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
}
```

```
//face en haut
```

```
if(getValue(image, i+1, j, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
}
```

```
//face en bas
```

```
if(getValue(image, i-1, j, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
}
```

//face derrière

```
if(getValue(image, i, j, k+1, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
}
```

//face devant

```
if(getValue(image, i, j, k-1, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```

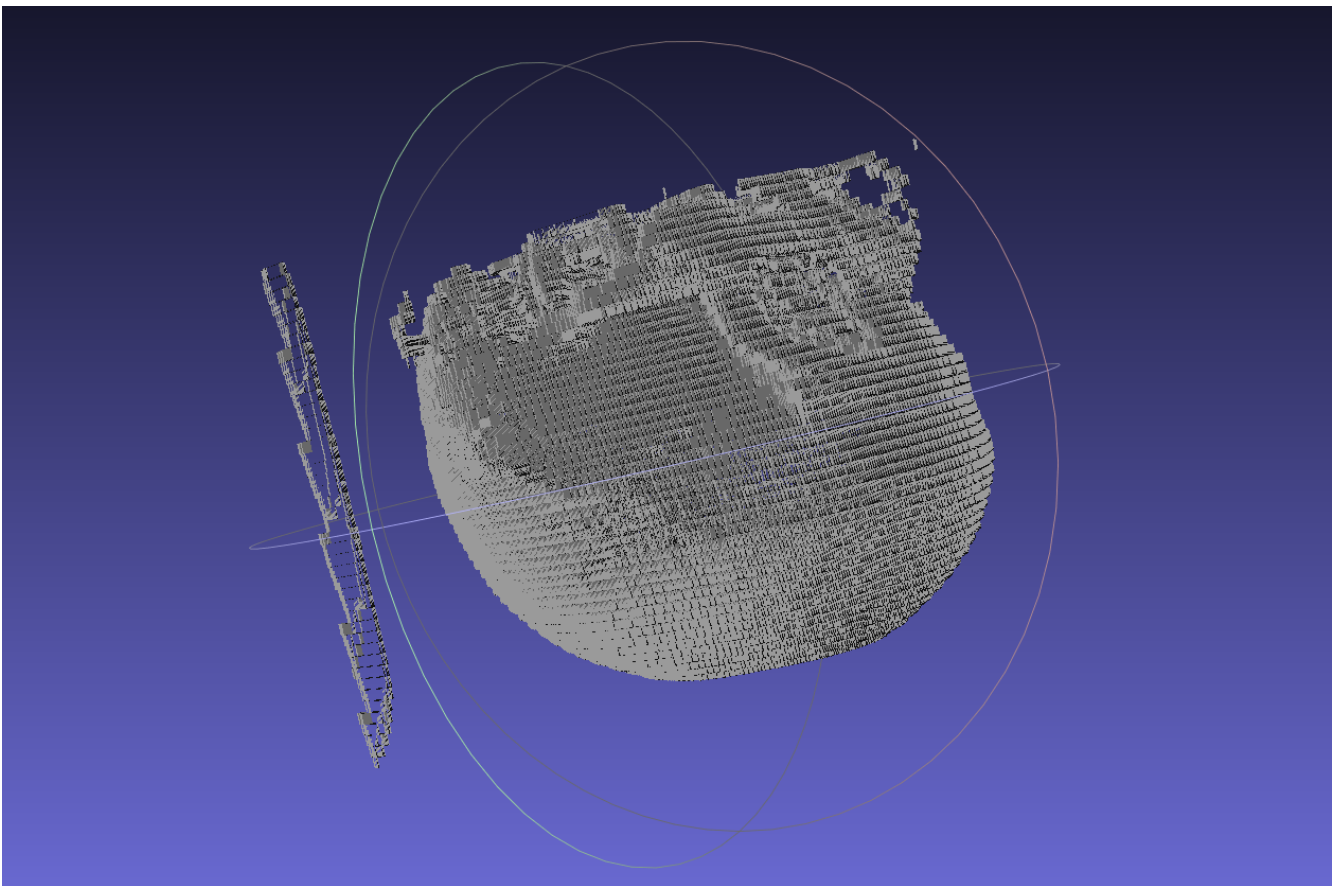
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
}
}

}
}
}
}

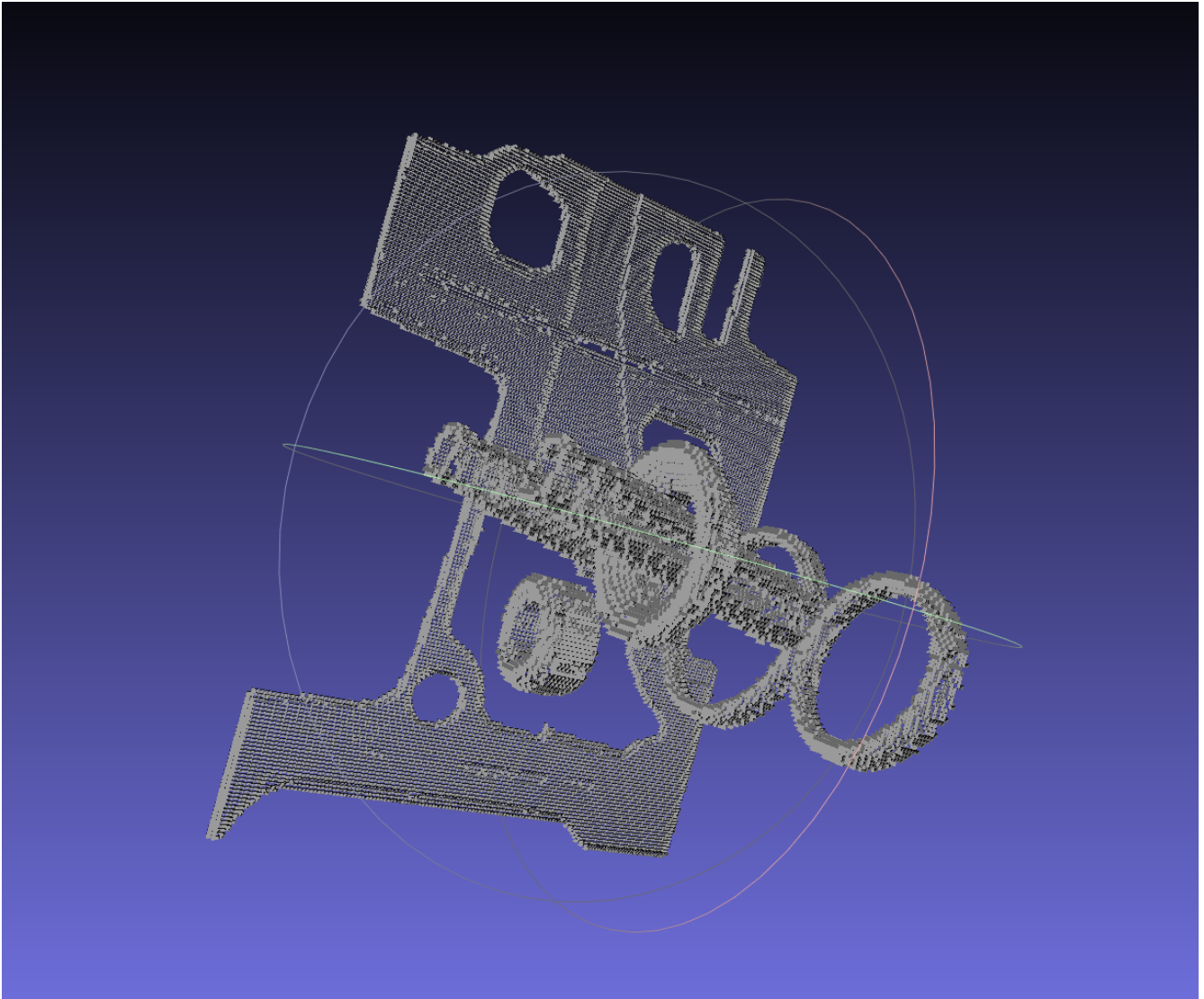
```

on ne crée pas deux fois les mêmes triangles entre deux voxels car

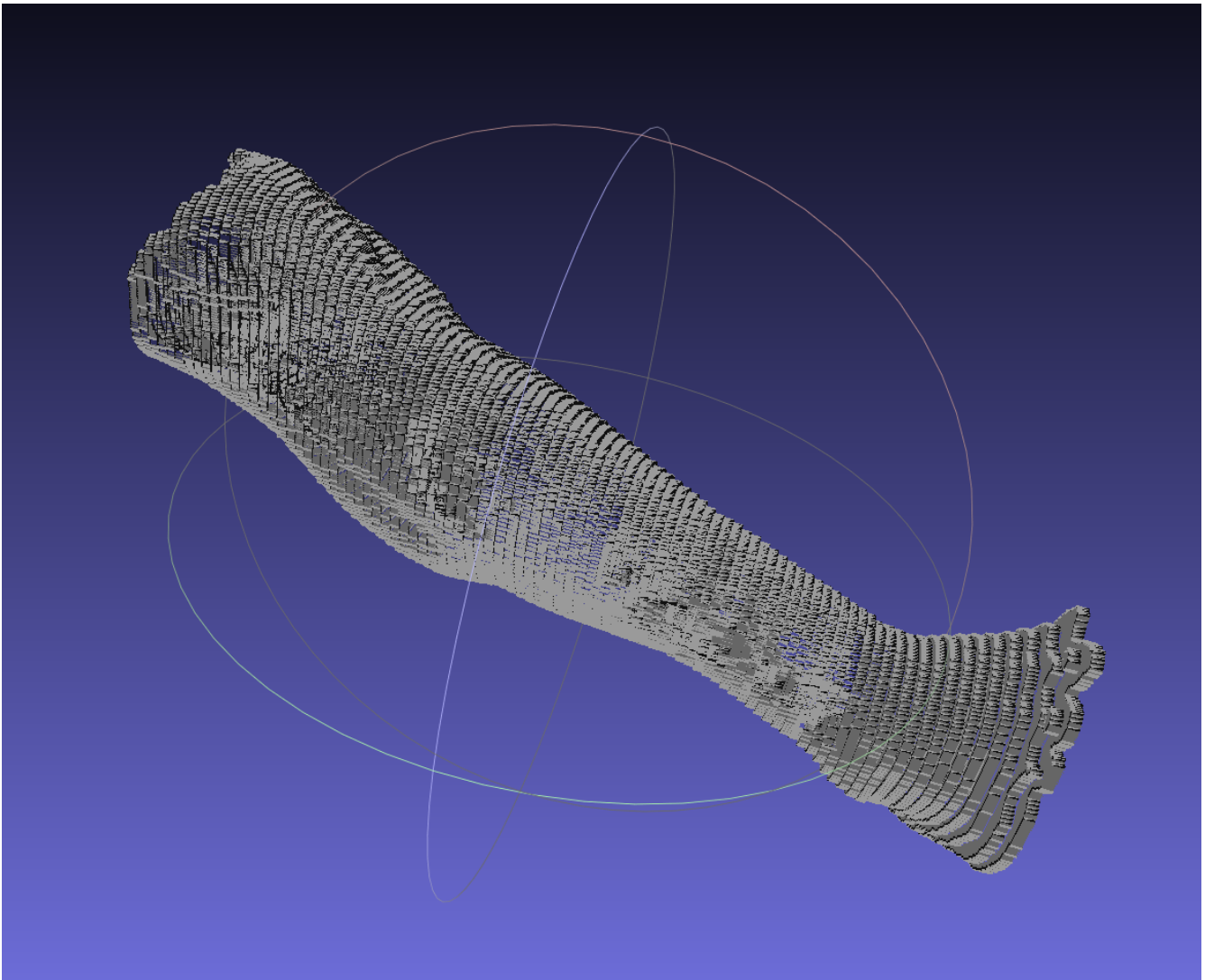
2.Le résultat après tests sur les images 3D:



manix avec seuil = 1100



engine avec seuil = 200



statut avec seuil = 50

3.Code:

```
#include <stdio.h>
#include <stdlib.h>

#define allocation_tableau(nom, type, nombre) \
if( (nom = (type*) calloc (nombre, sizeof(type) ) ) == NULL ) \
{ \
printf("\n Allocation dynamique impossible pour un pointeur-tableau \n"); \
exit(1); \
}

void lecture_image(char nomFichier[], unsigned short* image, int dimX, int dimY, int dimZ){

FILE* f;
unsigned short chaine;
```

```
if((f = fopen(nomFichier, "rb")) == NULL){  
    printf("pas d'accès en lecture au fichier");  
}else{
```

```
    int taille = dimX * dimY * dimZ;
```

```
    if( (fread((unsigned short*)image, sizeof(unsigned short), taille, f))  
        != (size_t) taille)  
    {  
        printf("\nErreur de lecture de l'image %s \n", nomFichier);  
        exit(EXIT_FAILURE);  
    }  
    fclose(f);
```

```
}
```

```
}
```

```
void ecrire_image(char nom_image[], unsigned short *image, int dimX, int dimY, int dimZ)  
{  
    FILE *f_image;  
    int taille_image = dimX * dimY * dimZ;
```

```
    if( (f_image = fopen(nom_image, "wb")) == NULL)  
    {  
        printf("\nPas d'accès en écriture sur l'image %s \n", nom_image);  
        exit(EXIT_FAILURE);  
    }  
    else  
    {
```

```
        if( (fwrite((unsigned short*)image, sizeof(unsigned short), taille_image, f_image))  
            != (size_t) taille_image)  
        {  
            printf("\nErreur de lecture de l'image %s \n", nom_image);  
            exit(EXIT_FAILURE);  
        }  
        fclose(f_image);  
    }  
}
```

```
unsigned short getValue(unsigned short* image, int i, int j, int k, int dimX, int dimY, int dimZ){  
    return image[k * dimX * dimY + (((dimY-1)-j)*dimX + i)];  
}
```

```
void seuillage(unsigned short* image, unsigned short seuil, int dimX, int dimY, int dimZ, float
sizeX, float sizeY, float sizeZ, char nomFich[]){
```

```
FILE* fichier;
```

```
unsigned short voxel =0;
```

```
fichier = fopen(nomFich,"w");
```

```
fprintf(fichier,"solid name\n");
```

```
for(int i=1; i<dimX-1; i++){
```

```
for(int j=1; j<dimY-1; j++){
```

```
for(int k=1; k<dimZ-1; k++){
```

```
//printf("(%d,%d,%d) = %hu\n",i,j,k, getValue(image,i,j,k,dimX,dimY,dimZ));
```

```
if(getValue(image, i, j, k, dimX, dimY, dimZ) >= seuil){
```

```
//face à gauche
```

```
if(getValue(image, i, j-1, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"endloop\n");
```

```
fprintf(fichier,"endfacet\n");
```

```
}
```

```
//face à droite
```

```
if(getValue(image, i, j+1, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
```

```
fprintf(fichier,"outer loop\n");
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
```



```
fprintf(fichier,"endloop\n");  
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");  
fprintf(fichier,"outer loop\n");  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"endloop\n");  
fprintf(fichier,"endfacet\n");  
}
```

```
//face en haut
```

```
if(getValue(image, i+1, j, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");  
fprintf(fichier,"outer loop\n");  
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"endloop\n");  
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");  
fprintf(fichier,"outer loop\n");  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);  
fprintf(fichier,"endloop\n");  
fprintf(fichier,"endfacet\n");  
}
```

```
//face en bas
```

```
if(getValue(image, i-1, j, k, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");  
fprintf(fichier,"outer loop\n");  
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"endloop\n");  
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");  
fprintf(fichier,"outer loop\n");  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);  
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
```

```
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
}
```

```
//face derrière
```

```
if(getValue(image, i, j, k+1, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j + 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
}
```

```
//face devant
```

```
if(getValue(image, i, j, k-1, dimX, dimY, dimZ) < seuil){
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
fprintf(fichier,"facet normal 0 0 0\n");
fprintf(fichier,"outer loop\n");
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i - 0.5)*sizeX, (j - 0.5)*sizeY, (k - 0.5)*sizeZ);
fprintf(fichier,"vertex %f %f %f\n", (i + 0.5)*sizeX, (j - 0.5)*sizeY, (k + 0.5)*sizeZ);
fprintf(fichier,"endloop\n");
fprintf(fichier,"endfacet\n");
```

```
}
```

```
}
```

```
}
```

```
}  
}  
}
```

```
int main(int argc, char* argv[])  
{  
    char cNomImgLue[250], cNomImgEcrire[250];  
    int dimX, dimY, dimZ;  
    float sizeX, sizeY, sizeZ;  
    unsigned short seuil;  
    if (argc != 10)  
    {  
        printf("Usage: nomfichier nomImageEcrire dimX dimY dimZ sizeX sizeY sizeZ seuil\n");  
        exit (1) ;  
    }  
    sscanf (argv[1], "%s", cNomImgLue);  
    sscanf (argv[2], "%s", cNomImgEcrire);  
    sscanf (argv[3], "%d", &dimX);  
    sscanf (argv[4], "%d", &dimY);  
    sscanf (argv[5], "%d", &dimZ);  
    sscanf (argv[6], "%f", &sizeX);  
    sscanf (argv[7], "%f", &sizeY);  
    sscanf (argv[8], "%f", &sizeZ);  
    sscanf (argv[9], "%hu", &seuil);  
  
    unsigned short* image;  
    int taille = dimX*dimY*dimZ;  
  
    allocation_tableau(image, unsigned short, taille);  
    lecture_image(cNomImgLue, image, dimX, dimY, dimZ);  
  
    unsigned short* image2;  
  
    allocation_tableau(image2, unsigned short, taille);  
  
    for(int i=0; i<taille; i++){  
        image[i] = bswap_16(image[i]);  
    }  
  
    seuillage(image, seuil, dimX, dimY, dimZ, sizeX, sizeY, sizeZ, cNomImgEcrire);
```

```
return 1;  
}
```