Rendu TP Imagerie 3D

Yasmine Khodja

Première partie:

- Lecture de l'image:

Après allocation de mémoire dans le main, j'utilise la méthode **lecture_image** qui rends un tableau rempli de unsigned short de l'image.

```
#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);

}
}
```

- La méthode **getValue**:

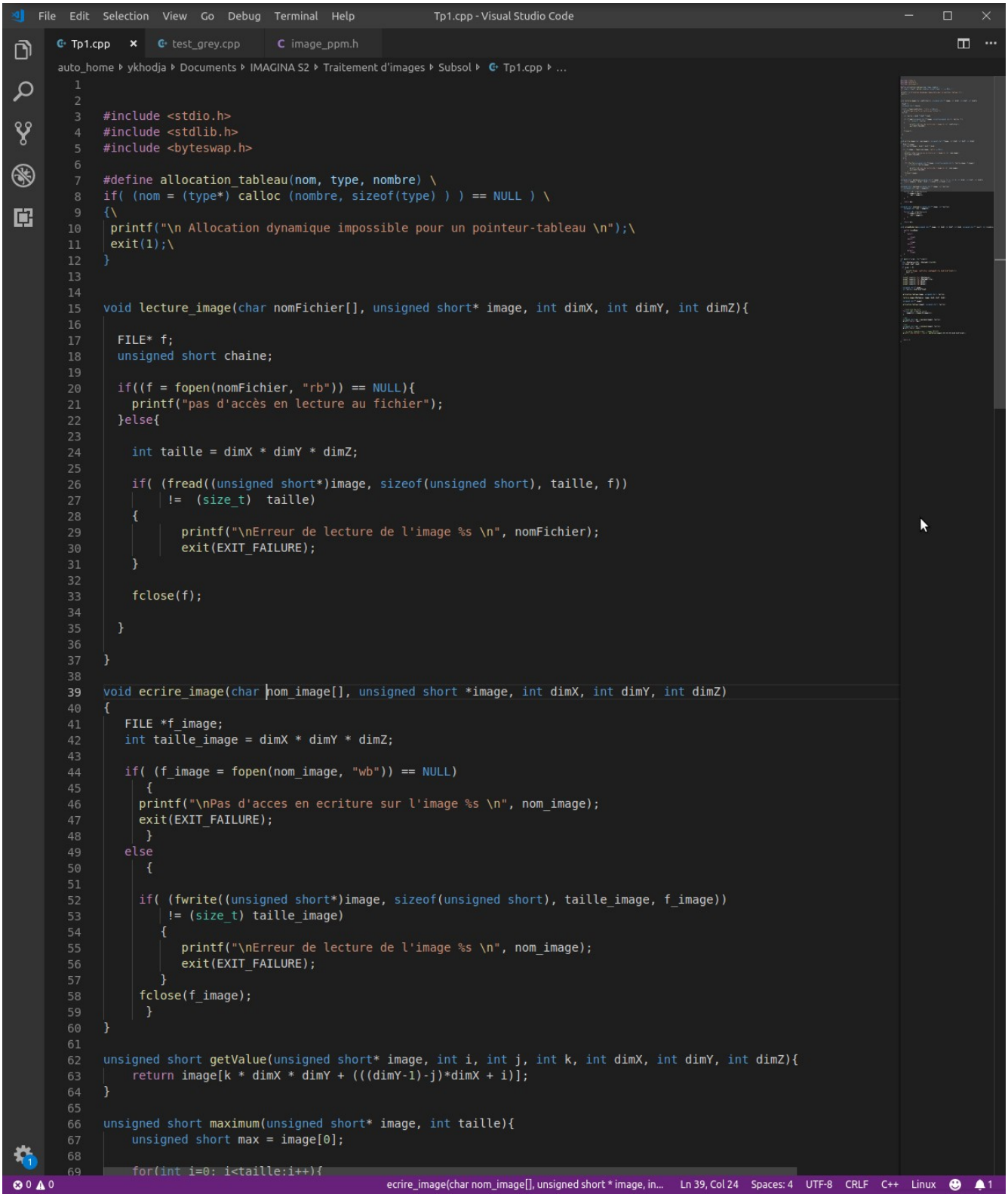
```
unsigned short getValue(unsigned short* image, int i, int j, int k, int dimX, int dimY, int dimZ){
return image[k * dimX * dimY + (((dimY-1)-j)*dimX + i)];
}
```

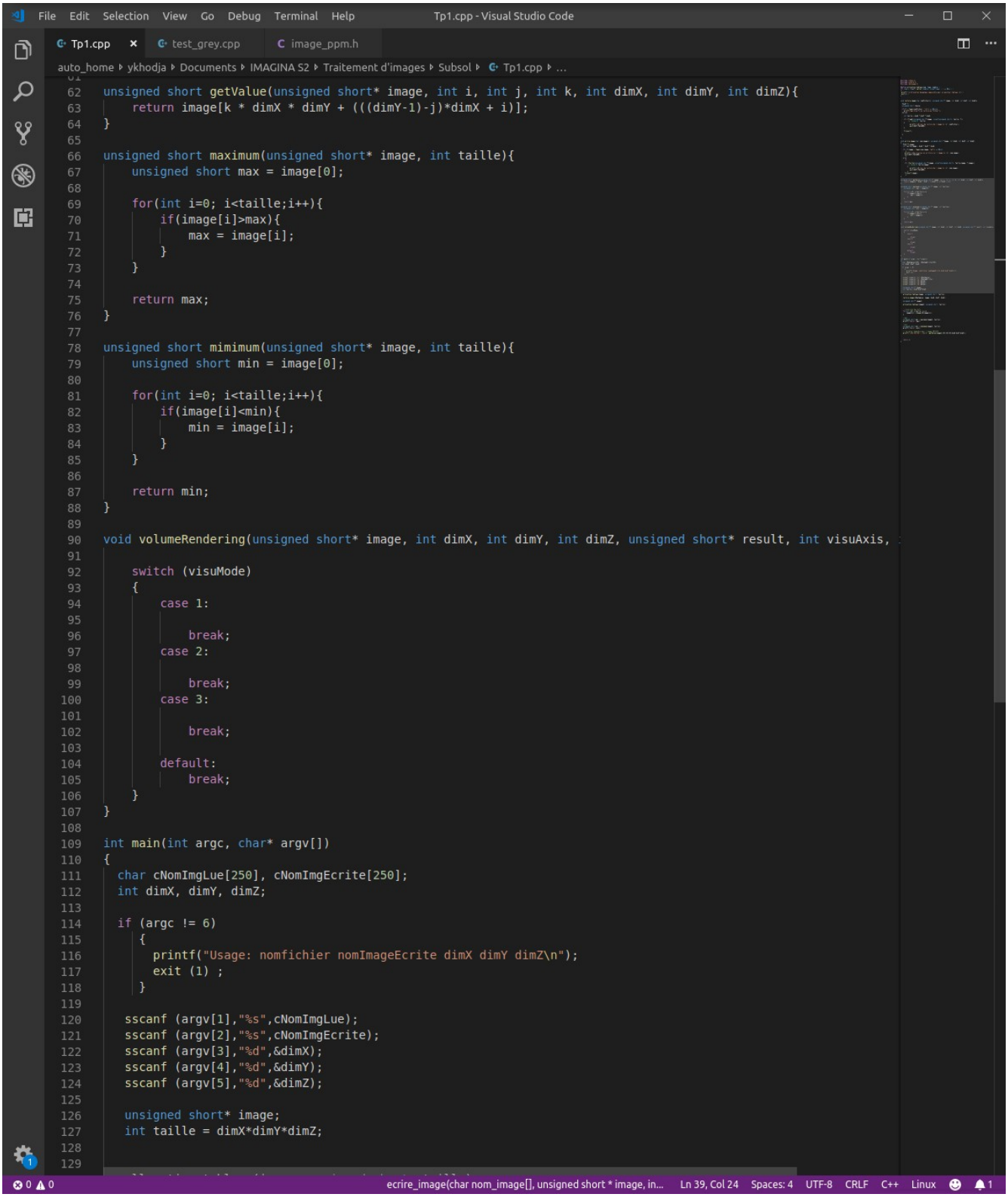
- La valeur minimale et maximale des voxels de l'image:

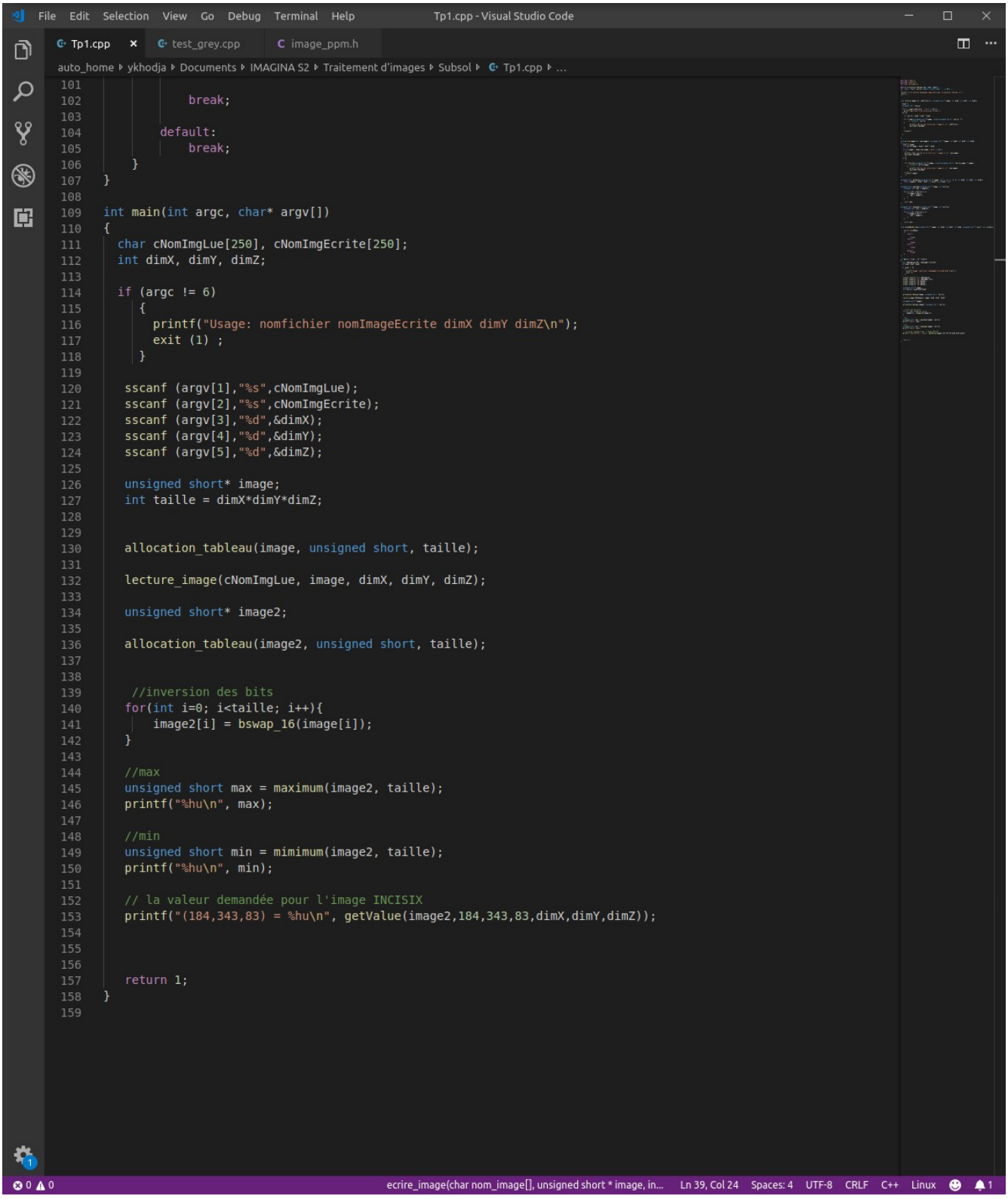
```
unsigned short maximum(unsigned short* image, int taille){
unsigned short max = image[0];

for(int i=0; i<taille;i++){
if(image[i]>max){
max = image[i];
}
}

return max;
}

unsigned short mimimum(unsigned short* image, int taille){
unsigned short min = image[0];

for(int i=0; i<taille;i++){
if(image[i]<min){
min = image[i];
}
}

return min;
}
```

- Pour afficher l'intensité d'un voxel, il suffit de faire appel à la fonction **getValue** avec les coordonnées donnés, exemple pour l'image INCISIX:

```
printf("(184,343,83) = %hu\n", getValue(image2,184,343,83,dimX,dimY,dimZ));
```


File Edit Selection View Go Debug Terminal Help
Tp1.cpp - Visual Studio Code
Tp1.cpp
test_grey.cpp
image_ppm.h
auto_home ▸ ykhodja ▸ Documents ▸ IMAGINA S2 ▸ Traitement d'images ▸ Subsol ▸ Tp1.cpp ▸ ...
#include <stdio.h>
#include <stdlib.h>
#include <byteswap.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'acces en ecriture 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)];
}
unsigned short maximum(unsigned short* image, int taille){
unsigned short max = image[0];
for(int i=0; i<taille;i++){
0 0
ecrire_image(char nom_image[], unsigned short * image, in...
Ln 39, Col 24
Spaces: 4
UTF-8
CRLF
C++
Linux
1


File Edit Selection View Go Debug Terminal Help
Tp1.cpp - Visual Studio Code
Tp1.cpp
test_grey.cpp
image_ppm.h
auto_home ▸ ykhodja ▸ Documents ▸ IMAGINA S2 ▸ Traitement d'images ▸ Subsol ▸ Tp1.cpp ▸ ...
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)];
}
unsigned short maximum(unsigned short* image, int taille){
unsigned short max = image[0];
for(int i=0; i<taille;i++){
if(image[i]>max){
max = image[i];
}
}
return max;
}
unsigned short mimimum(unsigned short* image, int taille){
unsigned short min = image[0];
for(int i=0; i<taille;i++){
if(image[i]<min){
min = image[i];
}
}
return min;
}
void volumeRendering(unsigned short* image, int dimX, int dimY, int dimZ, unsigned short* result, int visuAxis,
switch (visuMode)
{
case 1:
break;
case 2:
break;
case 3:
break;
default:
break;
}
}
int main(int argc, char* argv[])
{
char cNomImgLue[250], cNomImgEcrite[250];
int dimX, dimY, dimZ;
if (argc != 6)
{
printf("Usage: nomfichier nomImageEcrite dimX dimY dimZ\n");
exit (1) ;
}
sscanf (argv[1],"%s",cNomImgLue);
sscanf (argv[2],"%s",cNomImgEcrite);
sscanf (argv[3],"%d",&dimX);
sscanf (argv[4],"%d",&dimY);
sscanf (argv[5],"%d",&dimZ);
unsigned short* image;
int taille = dimX*dimY*dimZ;
0 0
ecrire_image(char nom_image[], unsigned short * image, in...
Ln 39, Col 24
Spaces: 4
UTF-8
CRLF
C++
Linux
1


File Edit Selection View Go Debug Terminal Help
Tp1.cpp - Visual Studio Code
Tp1.cpp
test_grey.cpp
image_ppm.h
auto_home ▸ ykhodja ▸ Documents ▸ IMAGINA S2 ▸ Traitement d'images ▸ Subsol ▸ Tp1.cpp ▸ ...
101 break;
102
103 default:
104 break;
105 }
106 }
107
108 int main(int argc, char* argv[])
109 {
110 char cNomImgLue[250], cNomImgEcrite[250];
111 int dimX, dimY, dimZ;
112
113 if (argc != 6)
114 {
115 printf("Usage: nomfichier nomImageEcrite dimX dimY dimZ\n");
116 exit (1) ;
117 }
118
119 sscanf (argv[1],"%s",cNomImgLue);
120 sscanf (argv[2],"%s",cNomImgEcrite);
121 sscanf (argv[3],"%d",&dimX);
122 sscanf (argv[4],"%d",&dimY);
123 sscanf (argv[5],"%d",&dimZ);
124
125 unsigned short* image;
126 int taille = dimX*dimY*dimZ;
127
128
129 allocation_tableau(image, unsigned short, taille);
130
131 lecture_image(cNomImgLue, image, dimX, dimY, dimZ);
132
133 unsigned short* image2;
134
135 allocation_tableau(image2, unsigned short, taille);
136
137
138 //inversion des bits
139 for(int i=0; i<taille; i++){
140 image2[i] = bswap_16(image[i]);
141 }
142
143 //max
144 unsigned short max = maximum(image2, taille);
145 printf("%hu\n", max);
146
147 //min
148 unsigned short min = mimimum(image2, taille);
149 printf("%hu\n", min);
150
151 // la valeur demandée pour l'image INCISIX
152 printf("(184,343,83) = %hu\n", getValue(image2,184,343,83,dimX,dimY,dimZ));
153
154
155
156 return 1;
157 }
158
159
0 0
ecrire_image(char nom_image[], unsigned short * image, in...
Ln 39, Col 24
Spaces: 4
UTF-8
CRLF
C++
Linux
1