

Rapport Tp1 donnée multimedia
Thomas Bergon

1) compiler le fichier => gcc test_grey.cpp

2) Seuillage noir et blanc

fichier : grey_2.cpp

J'ai mis le seuillage en 2 et 3 parties dans le même fichier il compte juste les arguments pour savoir lequel on souhaite utiliser.

Difficulté : aucune

```
if (argc <5)
{
    printf("Usage: ImageIn.pgm ImageOut.pgm Seuil1 Seuil2 [Seuil3] \n");
    exit (1) ;
}

sscanf (argv[1], "%s", cNomImgLue) ;
sscanf (argv[2], "%s", cNomImgEcrire);
sscanf (argv[3], "%d", &S1);
sscanf (argv[4], "%d", &S2);
if (argc==6){
    sscanf (argv[5], "%d", &S3);
}
```

2 parties :

```
if(argc==5){
    for (int i=0; i < nH; i++)
        for (int j=0; j < nW; j++)
        {
            if ( ImgIn[i*nW+j] < S1) ImgOut[i*nW+j]=0;
            else if (ImgIn[i*nW+j] < S2) ImgOut[i*nW+j]=128;
            else ImgOut[i*nW+j]=255;
        }
}
```

3 parties :

```
else if (argc==6){
    for (int i=0; i < nH; i++)
        for (int j=0; j < nW; j++)
        {
            if ( ImgIn[i*nW+j] < S1) ImgOut[i*nW+j]=0;
            else if (ImgIn[i*nW+j] < S2) ImgOut[i*nW+j]=90;
            else if (ImgIn[i*nW+j] < S3) ImgOut[i*nW+j]=180;
            else ImgOut[i*nW+j]=255;
        }
}
```

2_Lena.pgm 50 145

50 145 210



3) Histogramme :

fichier :histo.cpp

Difficulté :

- se souvenir du fonctionnement de fstream, installer gnuplot (brew install gnuplot)
- Fstream ne marche pas avec ma version de gcc à cause de l'architecture du m1 de Apple (arm64) il faut donc que j'utilise g++ qui lui marche = grosse perte de temps.
- Aussi un ctrl+C du fichier d'instruction donne :

```
plot "histo.dat" with lines  
undefined variable: "histo"
```

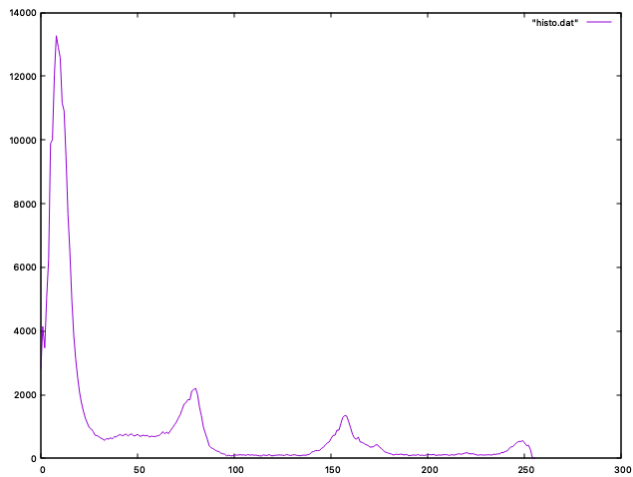
Mauvais guillemet. Avec les bons :

```
plot "histo.dat" with lines  
onts: Populating font family aliases took 172 ms.
```

Code :

```
int tab[256]={0};  
  
for (int i=0; i < nH; i++)  
    for (int j=0; j < nW; j++)  
    {  
        tab[ImgIn[i*nW+j]]+=1;  
    }  
  
free(ImgIn);  
std::ofstream myfile;  
myfile.open(fichierOut);  
for(int i=0;i<256;i++){  
    myfile<<i<<" "<<tab[i]<<std::endl;  
}  
myfile.close();
```

Résultat de ./a.out 02.pgm histo.dat
Visualiser avec Gnuplot :



4)

profil.cpp

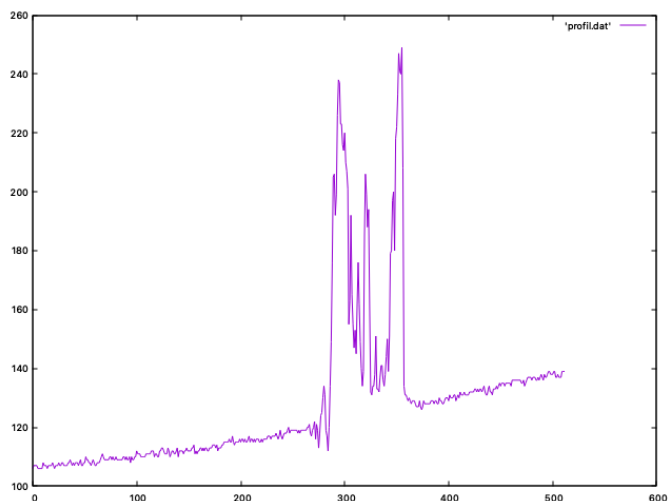
Difficultés :

Aucune

Si uniquement 4 arguments (pas de fichier de sortie) on affiche juste à l'écran sinon on met dans le fichier indiqué. Ligne = 1 , Colonne = 2 .

```
if(argc==4){
if(lc==1){
    if(num>nW){std::cout<<"num trop grand"<<std::endl;exit(1);}
    int i=num;
    for(int j=0;j<nH;j++){
        std::cout<<j<<" "<<int(ImgIn[i*nW+j])<<std::endl;
    }
}
else if(lc==2){
    if(num>nH){std::cout<<"num trop grand"<<std::endl;exit(1);}
    int j=num;
    for(int i=0;i<nW;i++){
        std::cout<<i<<" "<<int(ImgIn[i*nW+j])<<std::endl;
    }
}
}
```

Visualiser avec Gnuplot ./a.out 1 66 profil.dat



6)

color.cpp

Difficulté :

- Mauvais ordre dans le document (6 avant le 5)

```
for (int i=0; i < nTaille3; i+=3)
{
    nR = ImgIn[i];
    nG = ImgIn[i+1];
    nB = ImgIn[i+2];
    if (nR < S_R) ImgOut[i]=0; else ImgOut[i]=255;
    if (nG < S_G) ImgOut[i+1]=0; else ImgOut[i+1]=255;
    if (nB < S_B) ImgOut[i+2]=0; else ImgOut[i+2]=255;
}
```

./a.out lena.ppm test.ppm 60 80 140

Résultat :



5)

plot_color.cpp

Difficulté :

- Pas de documentation Gnuplot clair sur internet ...
- fstream mais résolut plutôt

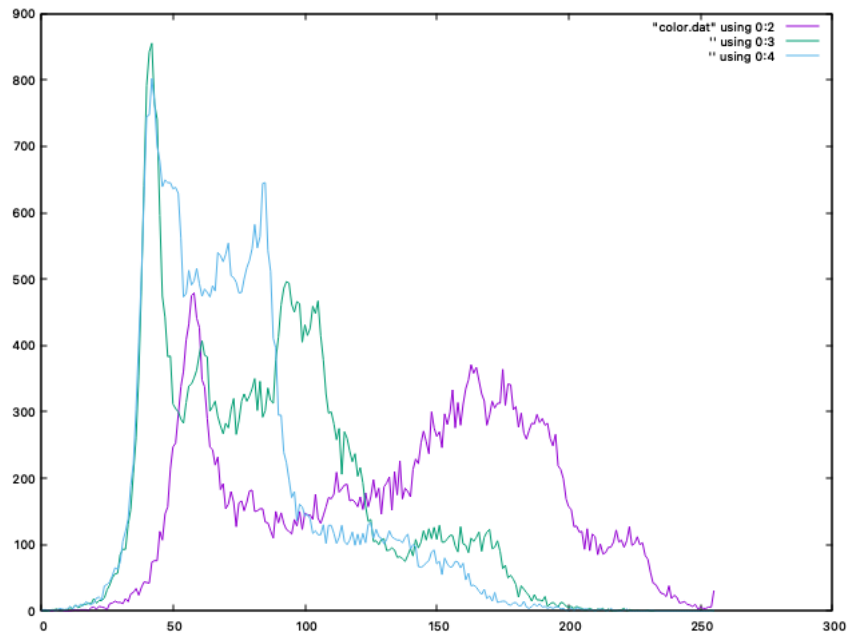
```
std::ofstream color;
color.open("color.dat");
int tabR[256]={0};
int tabG[256]={0};
int tabB[256]={0};

for (int i=0; i < nTaille3; i+=3)
{
    tabR[ImgIn[i]]+=1;
    tabG[ImgIn[i+1]]+=1;
    tabB[ImgIn[i+2]]+=1;
}

for(int i=0;i<256;i++){
    color<<i<<" "<<tabR[i]<<" "<<tabG[i]<<" "<<tabB[i]<<std::endl;
}
```

Résultat de ./a.out lena.ppm

Gnuplot> plot "color.dat" using 0:2 with lines, "" using 0:3 with lines, "" using 0:4 with lines



BONUS :

Otsu)

Otsu.cpp

Difficulté :

- La ligne avec les $q_1(t)$ qui en fait donne juste q_1
- Oublie d'initialiser les variables à 0 => image noir seuil énorme

Résultat :

`./a.out 01.pgm test.pgm`



Seuil infini)

infini_seuil.cpp

Difficulté : Aucune

Tableau de seuil.

```
if (argc <4)
{
    printf("Usage: ImageIn.pgm ImageOut.pgm seuil1 seuil2 ... \n");
    exit (1) ;
}

sscanf (argv[1],"%s",cNomImgLue) ;
sscanf (argv[2],"%s",cNomImgEcrit);

//tableau de la taille du nombre de Seuil = argc - nom_prog - imageIN - imageOUT
int nbSeuil=argc-3;
int Seuil[nbSeuil];
for(int i=0;i<nbSeuil;i++) {
    Seuil[i]=atoi(argv[3+i]);
}
```

```
//valeur a mettre si<Seuil
int tab[nbSeuil];
for(int i=0;i<nbSeuil;i++) tab[i]=i*floor(255/(nbSeuil-1));

for (int x=0; x < nH; x++){
    for (int y=0; y < nW; y++)
    {
        for(int i=0;i<nbSeuil;i++)
        {
            if ( ImgIn[x*nW+y] < Seuil[i]) {
                ImgOut[x*nW+y]=tab[i];
                i=nbSeuil;
            }
        }
    }
}
```

Fonction Interpolation)

Math_grey.cpp

```
#include <math.h>
int eq(int x,int S1,int S2,int number){
    printf("here \n");
    int t=floor((x-S1)/(S2-S1));
    if (number==1) return (int) pow(t,3)*255;
    if (number==2) return ((3*t*t) - (2*t*t*t))*255;
    else return (int) floor(sin(t*M_PI/2)*255);
}
```