



Analyse et traitement d'images

Compte-rendu TP1

Prise en main d'une librairie de traitement d'images

Louis Jean
Master 1 IMAGINE
Université de Montpellier
N° étudiant : 21914083

23 janvier 2023

Table des matières

1	Introduction	2
2	Seuillage d'une image au format pgm	3
3	Seuillage d'une image pgm avec plusieurs niveaux S1, S2, S3	4
3.1	Trois niveaux de gris	4
3.2	Quatre niveaux de gris	4
4	Profil d'une ligne ou d'une colonne d'une image pgm	5
5	Histogramme d'une image pgm	6
6	Histogrammes des 3 composantes d'une image couleur au format ppm	7
7	Seuillage d'une image couleur (ppm)	8
8	Seuillage automatique d'une image pgm	9

1 Introduction

Le but de ce TP est de faire une première prise en main de la bibliothèque de traitement d'images que nous utiliserons à chaque TP de cette UE. Tout au long de ce TP, nous utiliserons deux images de Lena, une au format pgm (de taille 512x512 pixels) et une au format ppm (de taille 200x200 pixels), que voici ci-dessous.



(a) Image de Lena au format pgm



(b) Image de Lena au format ppm

2 Seuillage d'une image au format pgm

Dans un premier temps, nous avons appliqué un simple seuil sur une image pgm.

Voici plusieurs exemples d'images seuillées avec des valeurs de seuil S différentes.

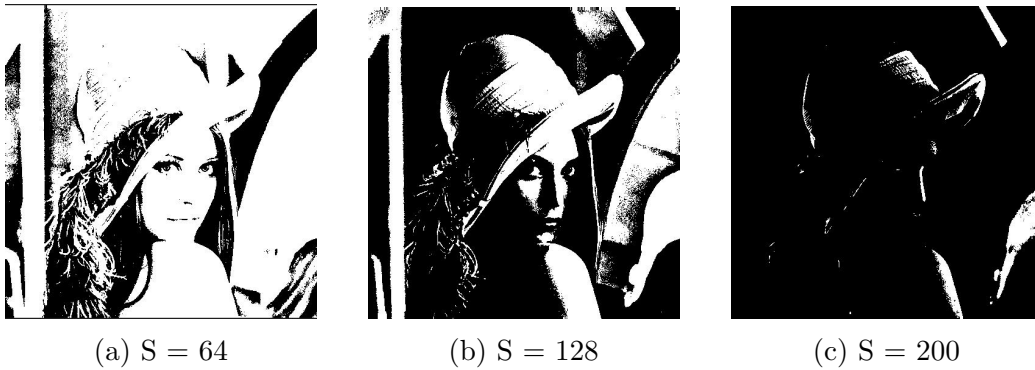


Figure 2: Images seuillées avec un seuil S différent

3 Seuillage d'une image pgm avec plusieurs niveaux S1, S2, S3

3.1 Trois niveaux de gris

Ensuite, nous avons étendu le seuillage à plusieurs niveaux, en commençant par utiliser deux seuils S1 et S2. J'ai choisi 0, 128 et 255 comme valeurs intermédiaires.

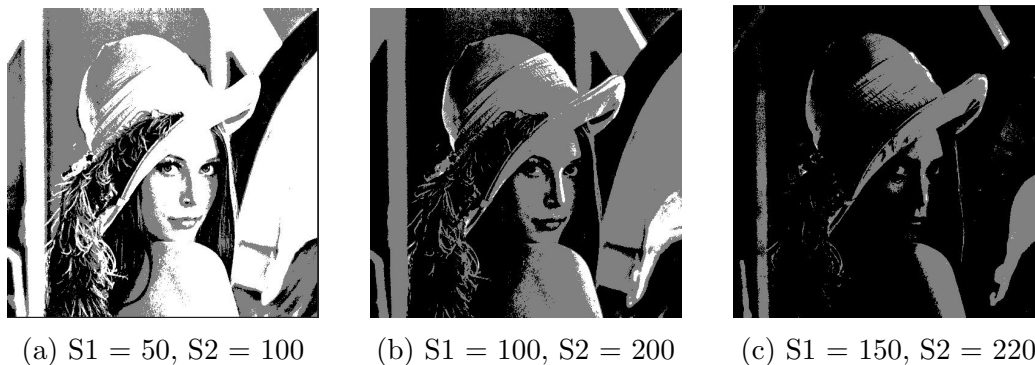


Figure 3: Images seuillées avec des seuils S1 et S2 différents

3.2 Quatre niveaux de gris

Ici, même principe mais avec trois seuils, S1, S2 et S3. J'ai choisi 0, 64, 128 et 255 comme valeurs intermédiaires.

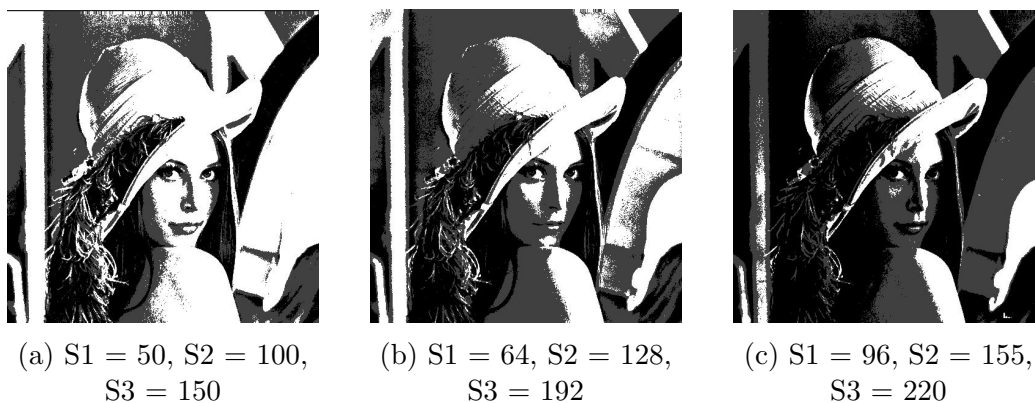


Figure 4: Images seuillées avec des seuils S1, S2 et S3 différents

4 Profil d'une ligne ou d'une colonne d'une image pgm

Le graphique de la figure ci-dessous présente le profil de la première colonne de l'image de Lena au format pgm, révélant les variations de clarté le long de celle-ci. L'axe des abscisses représente chaque pixel de la colonne (ici, on traite une image de 512x512 pixels), et l'axe des ordonnées répertorie la valeur (en niveau de gris) de ces pixels.

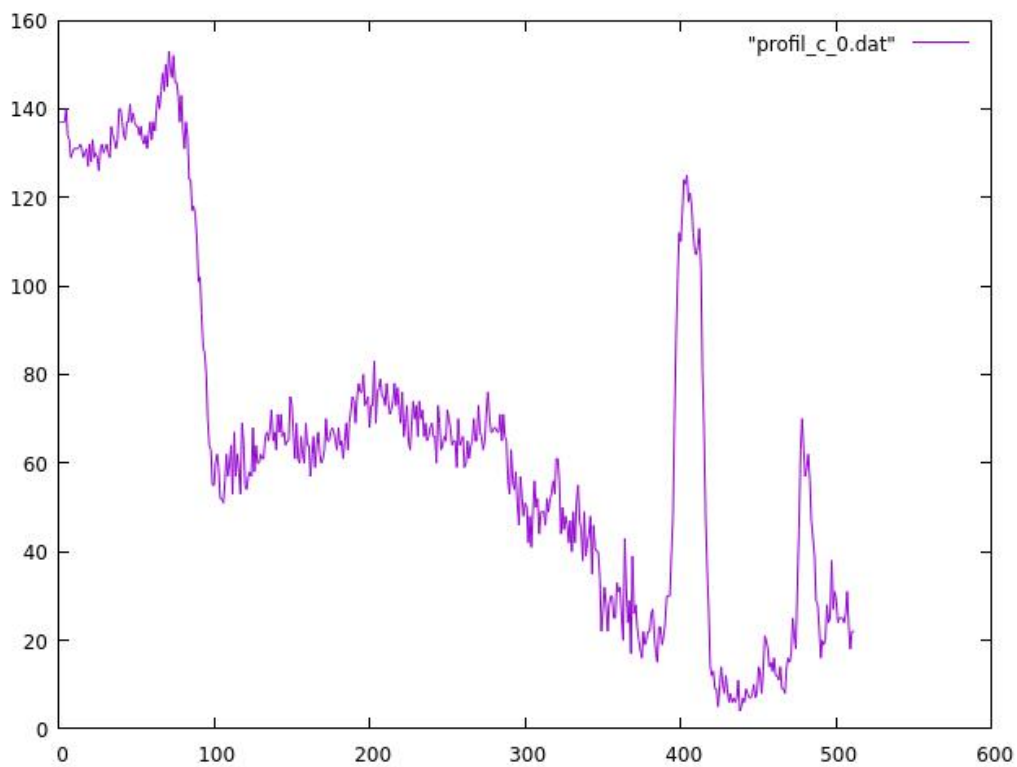


Figure 5: Profil de la colonne 0 de Lena.pgm

5 Histogramme d'une image pgm

La figure 6 représente l'histogramme de Lena.pgm, affichant la distribution des intensités des pixels. L'axe des ordonnées représente le nombre de pixels pour un niveau d'intensité spécifique, qui est indiqué sur l'axe des abscisses. Cela permet d'analyser le contraste de l'image et d'identifier les dominantes de clarté ou d'obscurité.

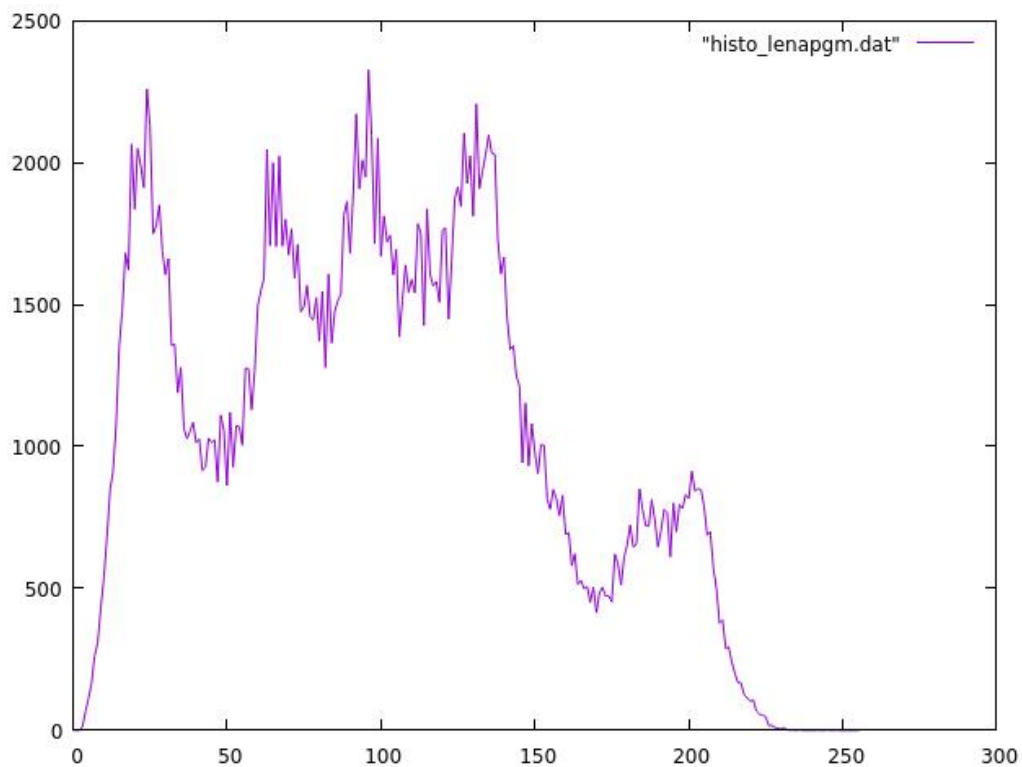


Figure 6: Histogramme de Lena.pgm

6 Histogrammes des 3 composantes d'une image couleur au format ppm

La figure 7 illustre les histogrammes des trois composantes couleur (rouge, vert, bleu) de l'image Lena.ppm. L'axe des ordonnées montre le nombre de pixels associés à chaque niveau d'intensité de couleur, représenté sur l'axe des abscisses. Cet histogramme révèle la distribution des couleurs dans l'image, permettant de discerner les dominantes de couleurs et le contraste global de l'image. Il fournit également des indices sur la luminosité et la saturation des couleurs présentes.

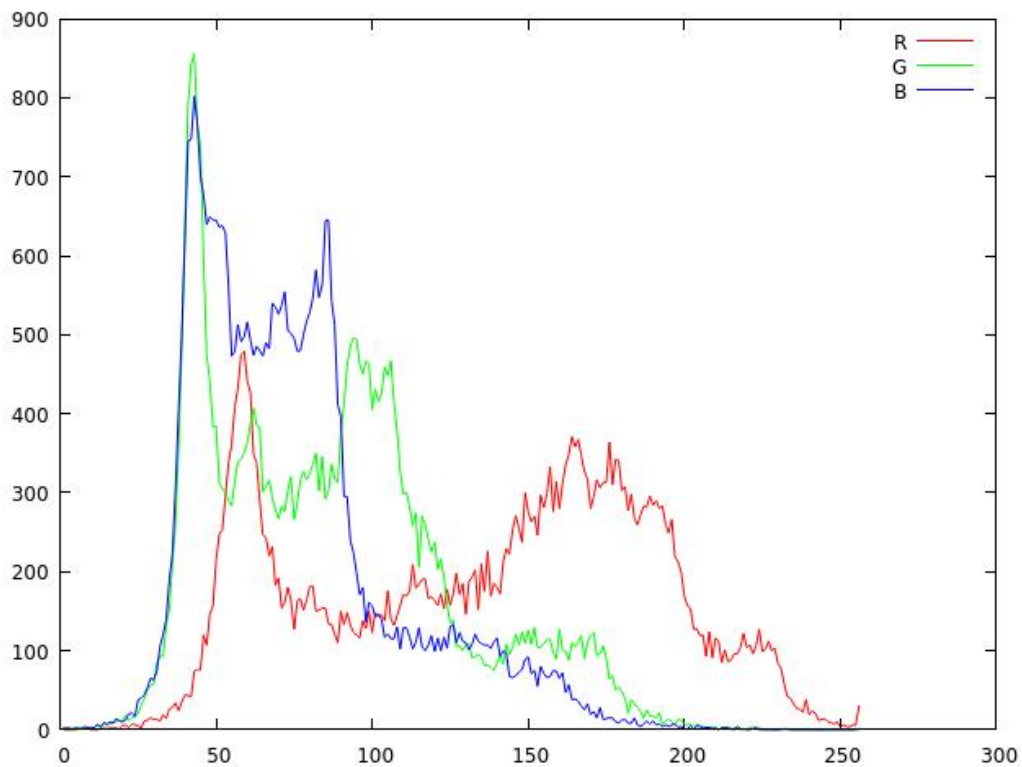


Figure 7: Histogramme de Lena.ppm

7 Seuillage d'une image couleur (ppm)

Pour seuiller une image couleur, nous avons seuillé une à une les composante couleur de l'image, en utilisant trois seuils S_R , S_G et S_B , comme l'illustre le code ci-dessous.

```
1 for(int i = 0; i < nTaille3; i+=3) {  
2     int nR = ImgIn[i];  
3     int nG = ImgIn[i+1];  
4     int nB = ImgIn[i+2];  
5     if(nR < S_R) ImgOut[i] = 0; else ImgOut[i] = 255;  
6     if(nG < S_G) ImgOut[i+1] = 0; else ImgOut[i+1] = 255;  
7     if(nB < S_B) ImgOut[i+2] = 0; else ImgOut[i+2] = 255;  
8 }
```

Listing 1: Seuillage de chaque composante couleur

Voici quelques exemples de résultats de ces seuillages.



(a) $S_R = S_G = S_B = 128$



(b) $S_R = 64, S_G = 96, S_B = 128$



(c) $S_R = 20, S_G = 100, S_B = 20$

Figure 8: Images seuillées avec des seuils S_R , S_G et S_B différents

8 Seuillage automatique d'une image pgm

Pour seuiller automatiquement une image pgm, je propose simplement de choisir comme valeur de seuil la moyenne des niveaux de gris des pixels de l'image considérée.

Voici le code qui calcule la moyenne des niveaux de gris d'une image pgm, pour l'utiliser comme seuil.

```
1 double moyenne = 0;
2 for(int i = 0; i < nH; i++) {
3     for(int j = 0; j < nW; j++) {
4         moyenne += ImgIn[i*nW+j];
5     }
6 }
7 moyenne /= nTaille;
8 moyenne = std::ceil(moyenne);
```

Listing 2: Calcul du niveau de gris moyen de l'image

Pour l'image de Lena, j'obtiens une moyenne de 100, et voici ce que cela donne comme résultat.



Figure 9: Image seuillée avec le seuil automatique

Merci pour le temps et l'attention que vous avez consacrés à la lecture de ce compte-rendu.