

# HAI804I – Analyse et Traitement d’Images

Fabien Caballero

February 9, 2023

## Contents

<b>1</b>	<b>Expansion dynamique</b>	<b>2</b>
1.1	BraZeLow.pgm . . . . .	2
1.2	black.ppm . . . . .	4
<b>2</b>	<b>Seuillage d’extrema des trois histogrammes</b>	<b>5</b>
<b>3</b>	<b>Egalisation d’histogramme</b>	<b>8</b>
<b>4</b>	<b>Spécification d’histogramme</b>	<b>11</b>

# 1 Expansion dynamique

## 1.1 BraZeLow.pgm



Figure 1: BraZeLow.pgm image d'origine

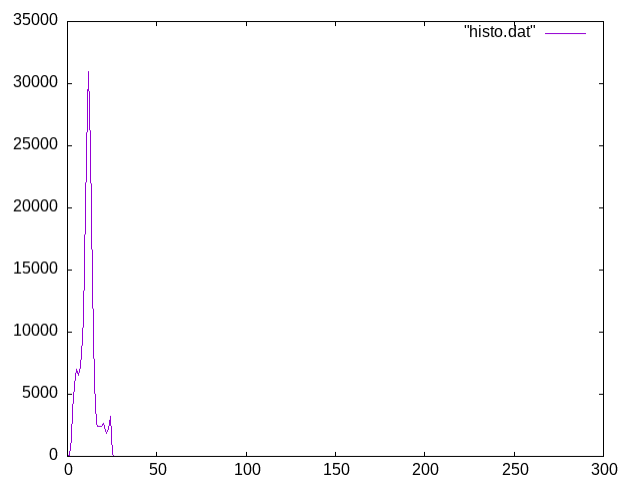


Figure 2: histogramme de BraZeLow.pgm

$\alpha = -10$   $\beta = 10$

Pour faire l'expansion de l'histogramme on cherche  $a_0$  et  $a_1$  tel que  $a_0$  soit la première couleur qui a une occurrence différente de 0 et  $a_1$  la dernière. Grâce à ça on calcule  $\alpha$  et  $\beta$  de la manière suivante:  
 $\alpha = (-255 \cdot a_0) / (a_1 - a_0)$   $\beta = 255 / (a_1 - a_0)$

Puis ensuite pour chaque pixel on attribue à notre pixel de sortie  $p = (-255 / (a_1 - a_0)) \cdot (a_0 - p)$ .

On voit donc apparaître l'animal inconnu, car l'image c'est éclaircie.

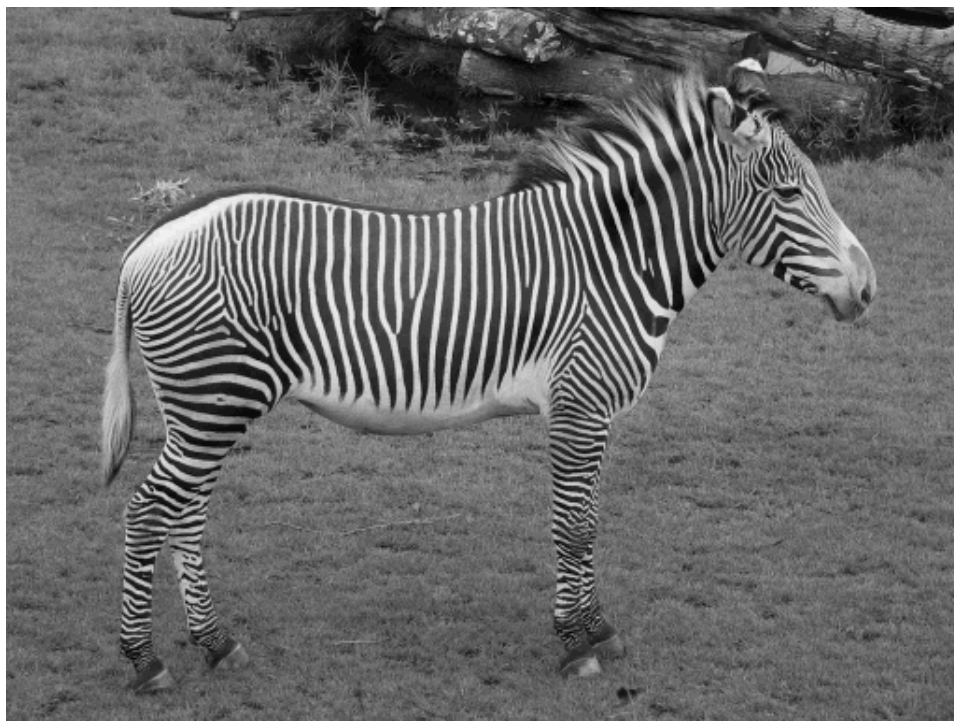
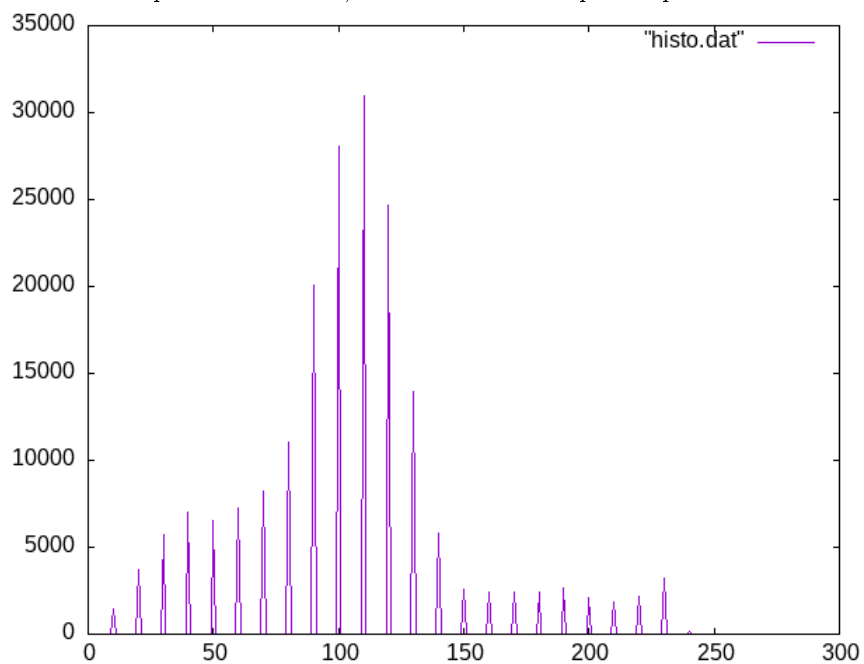


Figure 3: BraZeLow'.pgm

On remarque sur l'histogramme que certaines valeurs n'ont pas d'occurrences, car certaines ont disparues pour



devenir une autre plus claire.

Figure 4: histogramme BraZeLow'.pgm

## 1.2 black.ppm

On fait la même chose, qu'on applique sur chaque composantes (Rouge, Vert, Bleu)

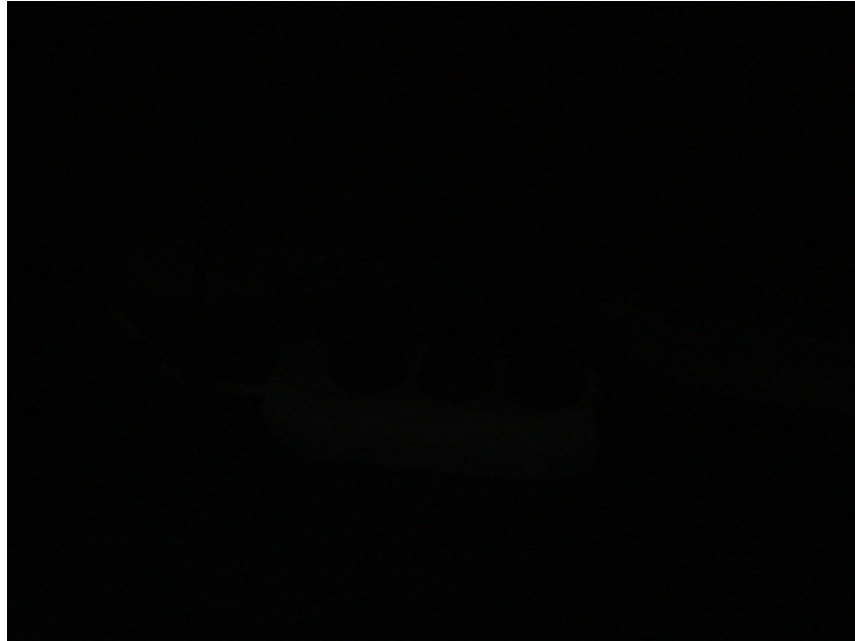


Figure 5: black.ppm



Figure 6: black'.ppm

```
alpha_Rouge= 0 beta_Rouge=19  
alpha_Vert= 0 beta_Vert=21  
alpha_Bleu= 0 beta_Bleu=21
```

## 2 Seuillage d'extrema des trois histogrammes

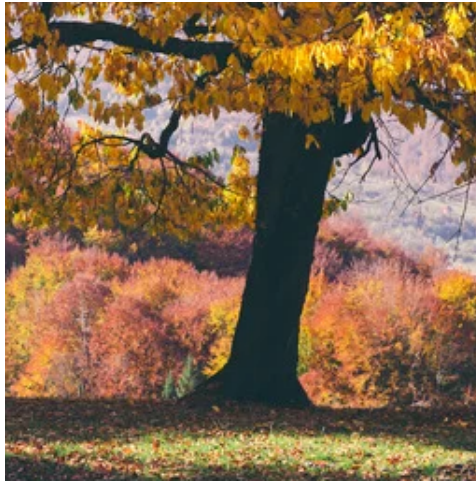


Figure 7: image originale

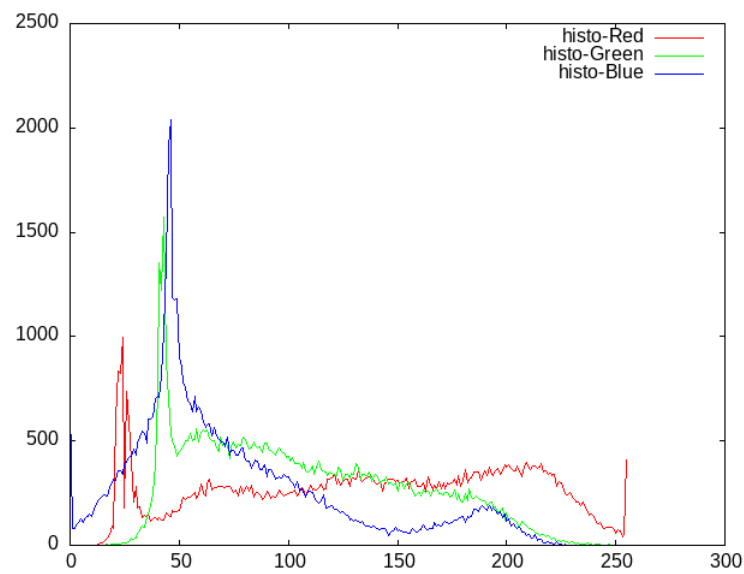


Figure 8: histogramme de l'image originale

En fonction de l'histogramme on choisi arbitrairement des valeurs de seuil et les couleurs inférieures à la couleur min sont changées par la couleur min pareil pour la couleur max. On fait ça pour chaque composante et on réassemble nos composantes.



Figure 9: image seuillée

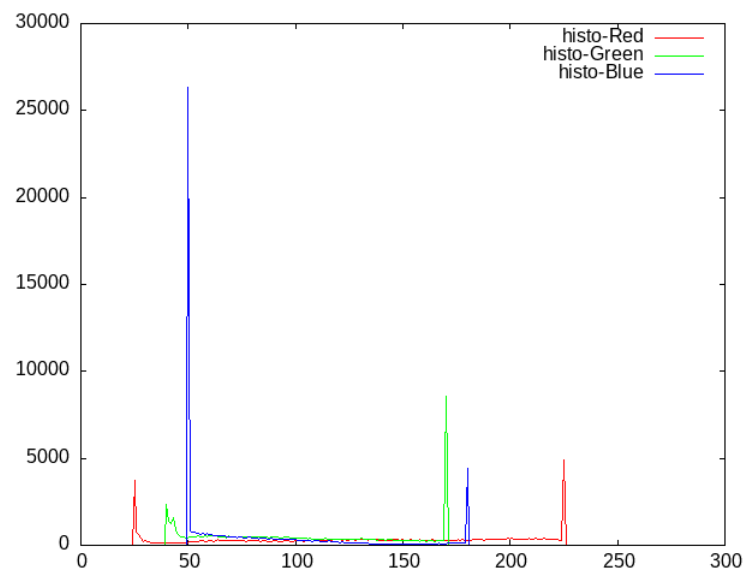


Figure 10: histogramme de l'image seuillée

On calcule la ddp en faisant l'histogramme et en divisant chaque occurrence par la taille de l'image.

Pour faire la fonction de répartition, il faut pour chaque valeur  $i$ , prendre la somme des probabilités des occurrences de 0 à  $i$ .

Pour égaliser une image on attribut à chaque pixel  $i$ , la valeur de la fonction de répartition pour la couleur originale de ce pixel et on multiplie par 255, pour avoir une couleur.

Même processus que précédemment

Puis on réalise l'expansion dynamique comme pour black.ppm avec cette fois-ci notre image seuillée.

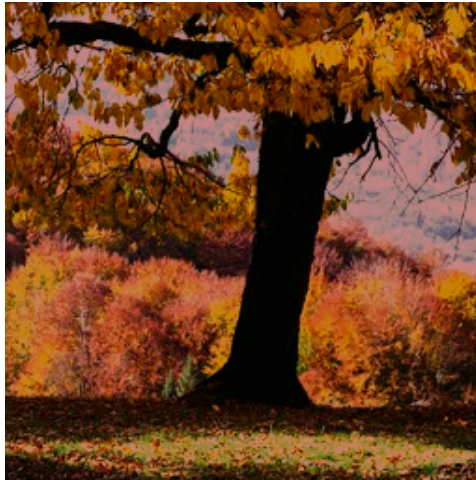


Figure 11: image finale

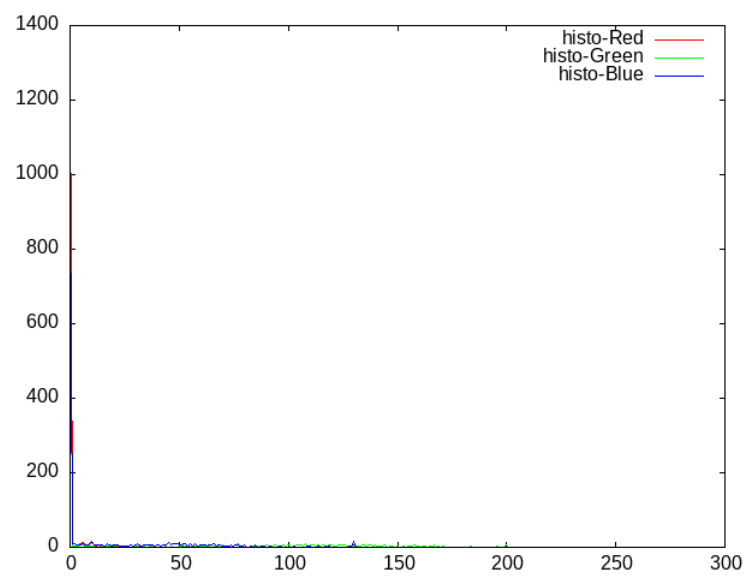


Figure 12: histogramme de l'image finale

### 3 Egalisation d'histogramme



Figure 13: image originale

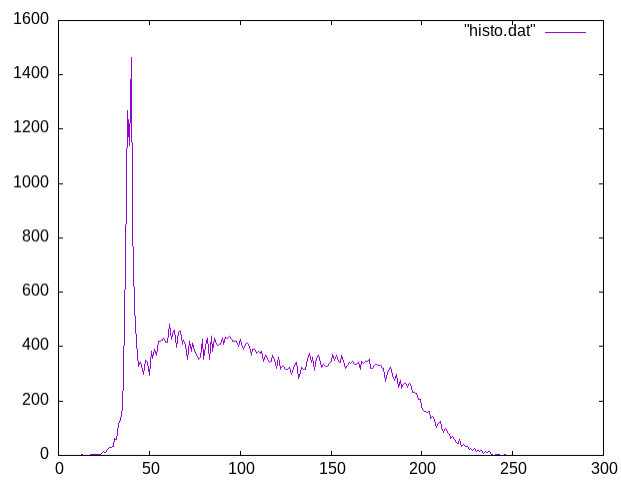


Figure 14: histogramme de l'image originale



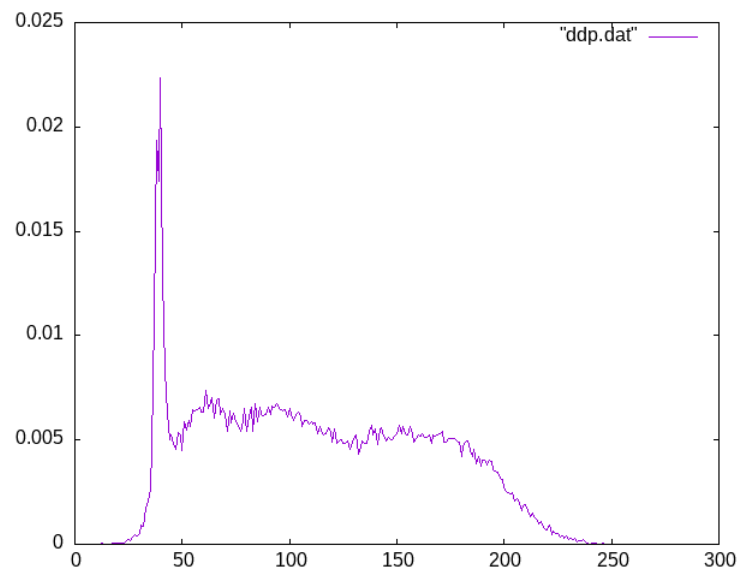


Figure 15: ddp de l'image originale

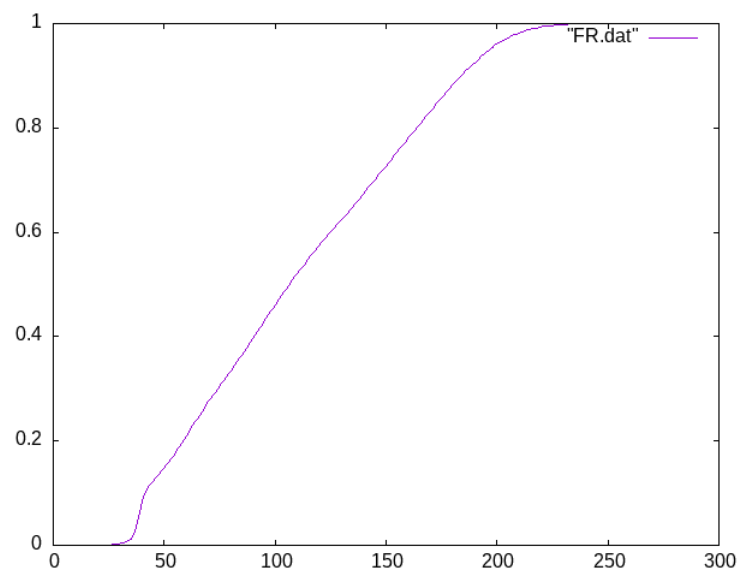


Figure 16: fonction de répartition de l'image originale



Figure 17: image égalisée

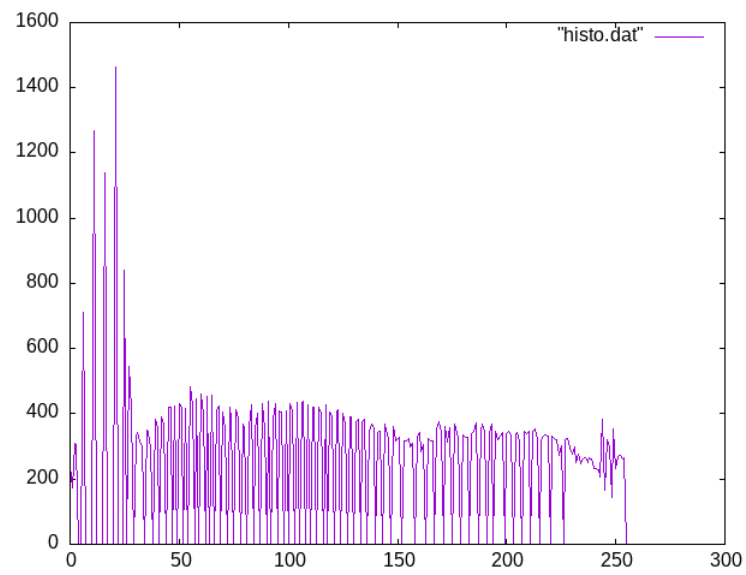


Figure 18: histogramme de l'image égalisée

## 4 Spécification d'histogramme

On prend une image cible A(ici lena.pgm)

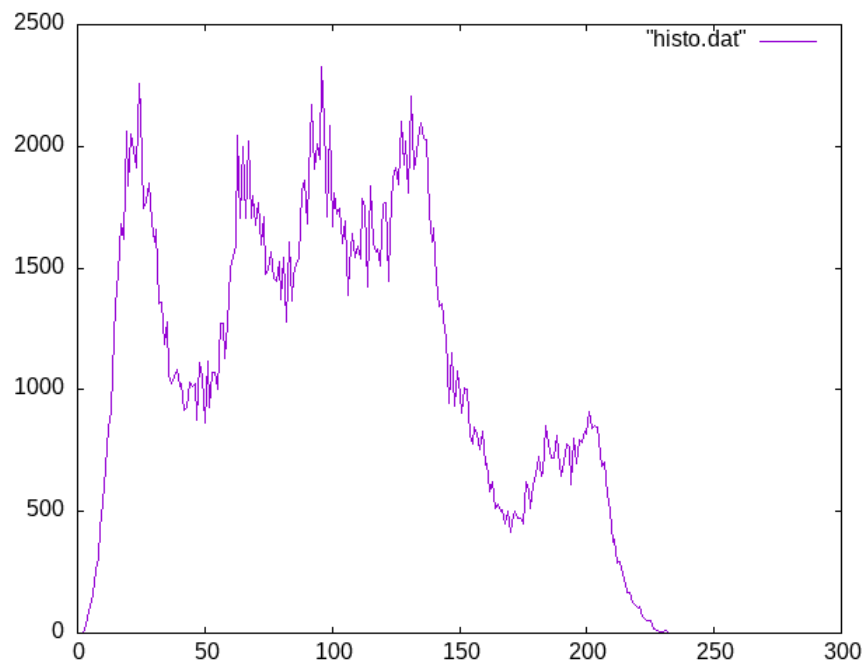


Figure 19: image cible

Ainsi qu'une image B qu'on vas spécifier. Et on va chercher à approcher l'histogramme de B vers celui de A.



Figure 20: image à spécifier

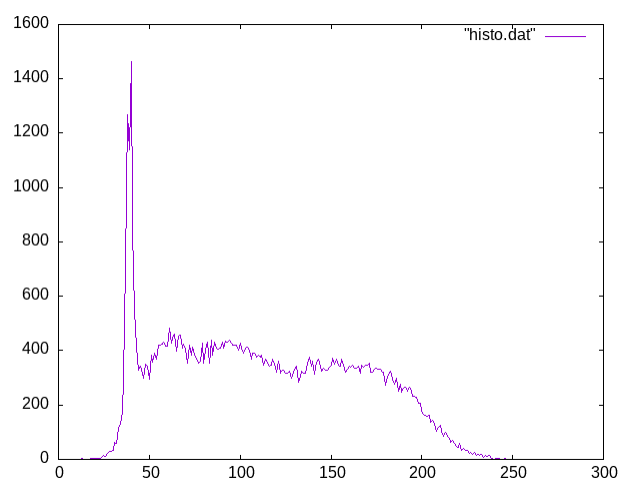


Figure 21: histogramme de l'image à spécifier



Figure 22: image à spécifier, égalisée

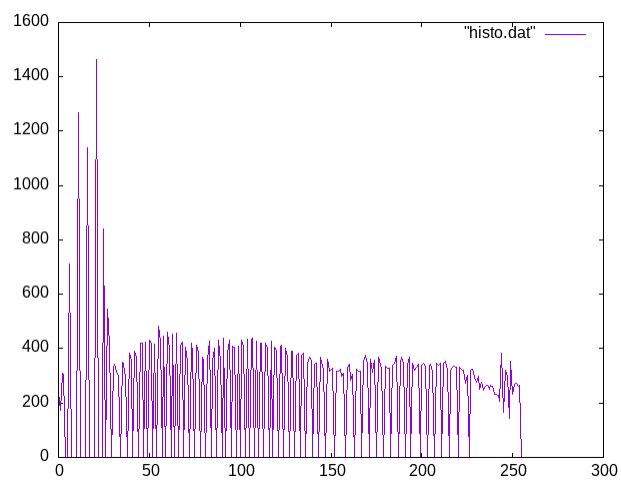


Figure 23: histogramme de l'image à spécifier, égalisée



Figure 24: image à spécifiée

Une fois égalisée on la spécifie pour cela on cherche le premier indice  $k$ , tel que la valeur  $k$  de la fonction de répartition de l'image A soit supérieure ou égale au pixel de l'image B égalisée. Une fois ce  $k$  trouvé on affecte ce  $k$  à la valeur du pixel de sortie. On fait ça pour tous les pixels et notre image est spécifiée. Et on remarque que l'histogramme de B suit la tendance de l'histogramme A.

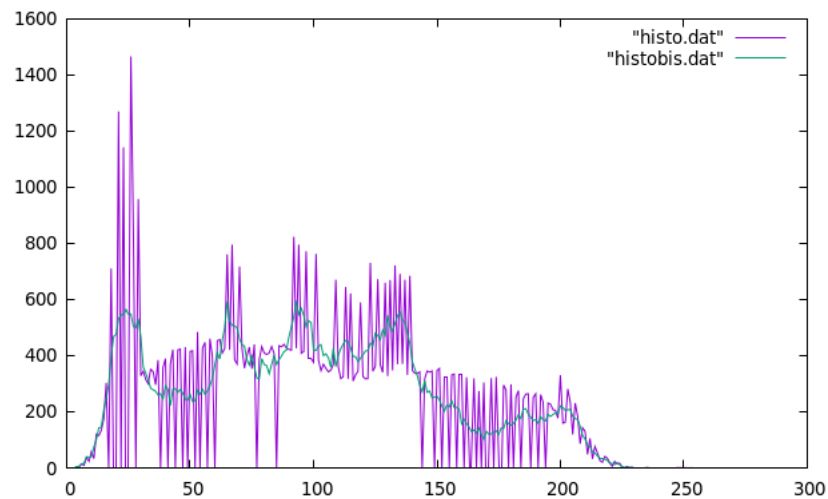


Figure 25: histogramme de l'image à spécifiée (en violet) et histogramme de lena égalisée (vert)