

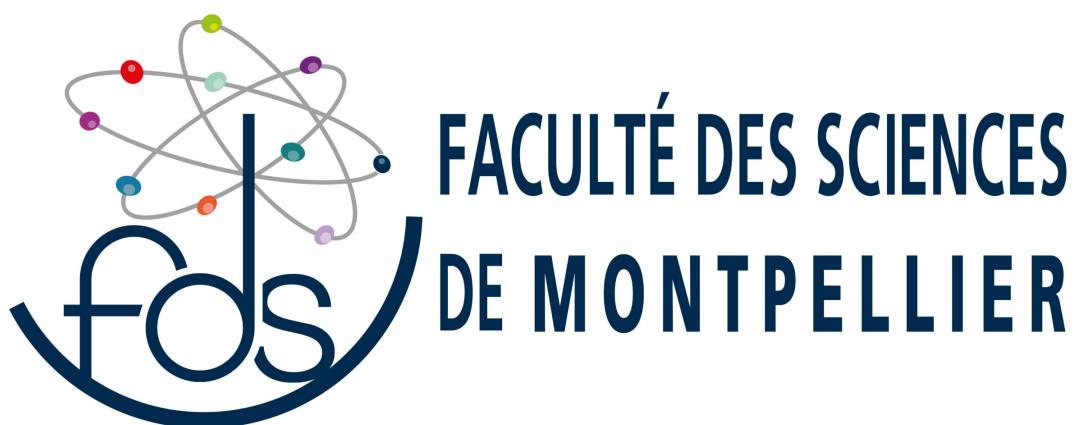
UNIVERSITÉ DE MONTPELLIER

M1 IMAGINE - Analyse et Traitement d'image TPn°3
Compte Rendu - Spécification des histogrammes

Etudiant :
Guillaume Bataille

Encadrant :
William PUECH

Année 2022-2023



0.1 Contexte et objectif

Lors du cours, nous avons vu l'importance et la pertinence du traitement des histogrammes pour modifier des aspects souhaités dans une image.

L'objectif de ce TP est d'effectuer des prétraitements sur une image à partir de méthodes de transformation d'histogramme. Voici les images non-fournies utilisées dans le tp :



FIGURE 1 – Araignée pour l'exercice de seuillage des extremas



FIGURE 2 – Arbre pour l'exercice de d'égalisation



FIGURE 3 – Tortue pour l'exercice de segmentation

Sommaire

0.1 Contexte et objectif	1
1 Expansion dynamique	3
2 Seuillage des extrems	5
3 Egalisation des histogramme	7
4 Spécification	9
5 Conclusion	13

1. Expansion dynamique

Certaines images souffre de ne pas assez bien couvrir le spectre des intensités de gris. Cela peut arriver sur des images trop claires ou trop sombre.

Pour y remédier, on décide d'étendre l'histogramme de ses images afin de l'étendre pour qu'il couvre tout le spectre en conservant sa forme générale.

Cela s'obtient en manipulant a_0 et a_1 de l'histogramme (les valeurs non nulles min et max de l'histogramme).

On en dégage les valeurs A et B :

$$A = ((-255 * a_0) / (a_1 - a_0))$$

$$B = (255 / (a_1 - a_0))$$

Enfin on interpole la nouvelle valeur du pixel p comme étant :

$$p' = (a_0 - p) * (-1 * (255 / (a_1 - a_0)))$$

En voici un exemple BraZeLow.pgm trop sombre et expanded avec **A = -10** et **B = 10** :

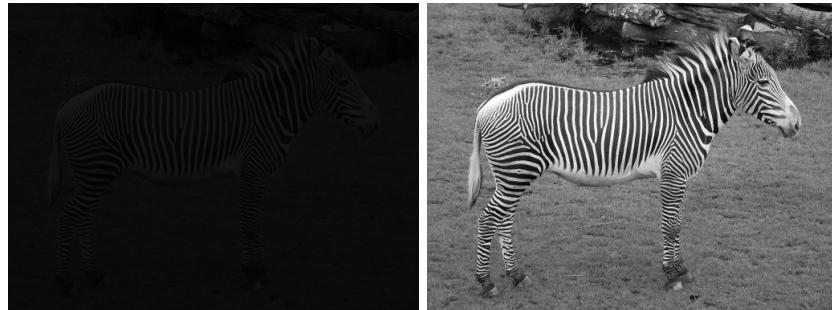


FIGURE 1.1 – BraZeLow.pgm original -> BraZeLow_expanded.pgm

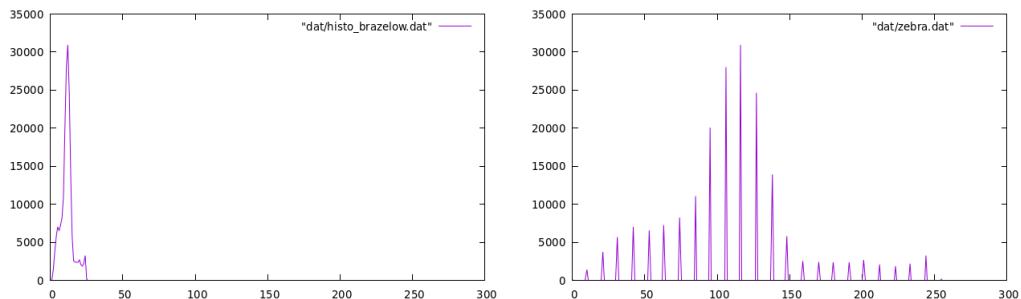


FIGURE 1.2 – Les histogrammes avant et après traitement

On constate que l'histogramme expanded est composé de pics. C'est normal parce qu'étendre un histogramme concentré sur un petit lot de valeur ne garanti pas de couvrir les 255 valeurs une fois expanded.

On se pose la même question avec une image RGB.

Cela revient juste à faire le traitement sur chaque composante de l'image.

Voici un exemple sur black.pgm :

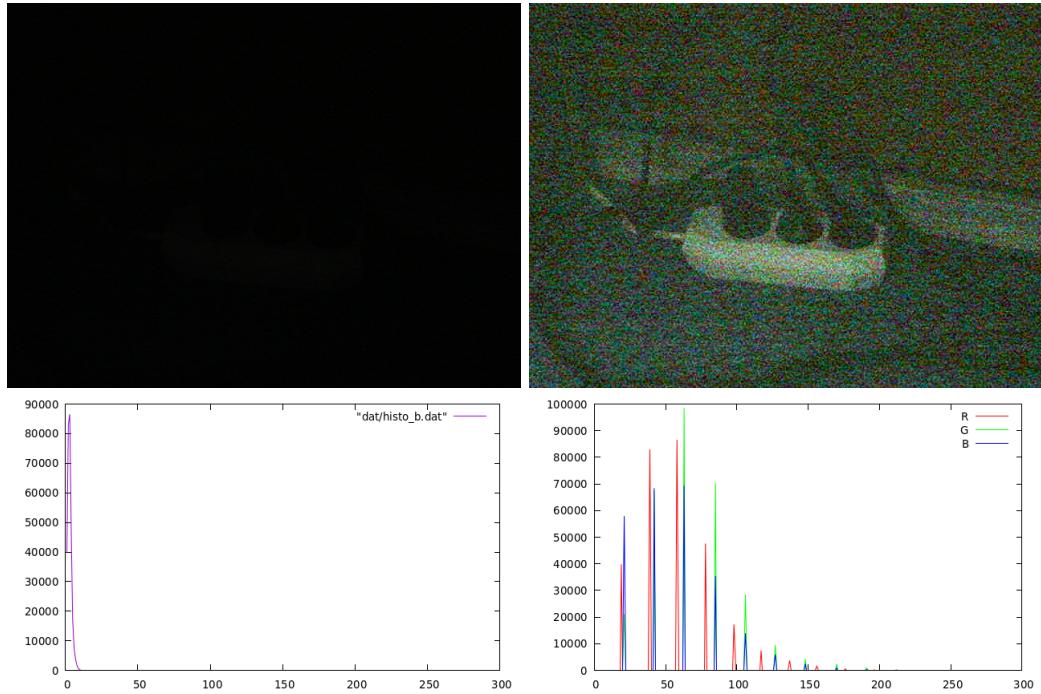


FIGURE 1.3 – black.ppm original, black_expanded.pgm et l'histogramme de black.ppm original

Rouge :

Borne min a0 :0

Borne min a1 :13

Valeur du coefficient alpha -> 0

Valeur du coefficient beta -> 19

Vert :

Borne min a0 :0

Borne min a1 :12

Valeur du coefficient alpha -> 0

Valeur du coefficient beta -> 21

Bleu :

Borne min a0 :0

Borne min a1 :12

Valeur du coefficient alpha -> 0

Valeur du coefficient beta -> 21

On constate un bruit sur l'image de couleur rouge verte et bleue. Cela s'explique par le fait que les composantes R G et B étant étendues indépendamment, elles ont de légères différences dans leurs nouvelles valeurs.

2. Seuillage des extremas

Cependant, on peut se faire duper si les valeurs a_0 et a_1 sont 0 et 255 mais que la majorité des pixels reste très sombre ou très clair. On voit donc qu'ici, les extremes faussent le traitement à faire.

On décide donc d'éliminer ses extremes en implémentant la version la plus naïve, donner un seuil min et max et écraser toute valeur hors seuil sur ces min et max.

On va travailler sur une image couleur. Tout d'abord on détermine les seuils min et max pour chaque composante à partir de l'histogramme. On choisit arbitrairement en regardant l'histogramme :

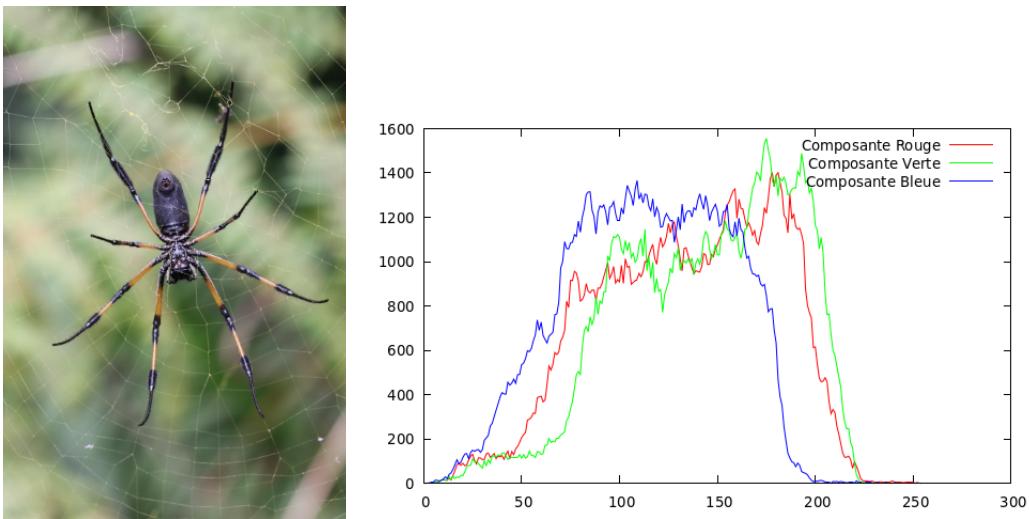


FIGURE 2.1 – Histogramme RGB de l'araignée

Pour R : min = 55 et max = 200

Pour G : min = 60 et max = 210

Pour B : min = 50 et max = 180

Maintenant, on peut seuiller l'image via la méthode évoqué précédemment, et ce sur chaque composante :

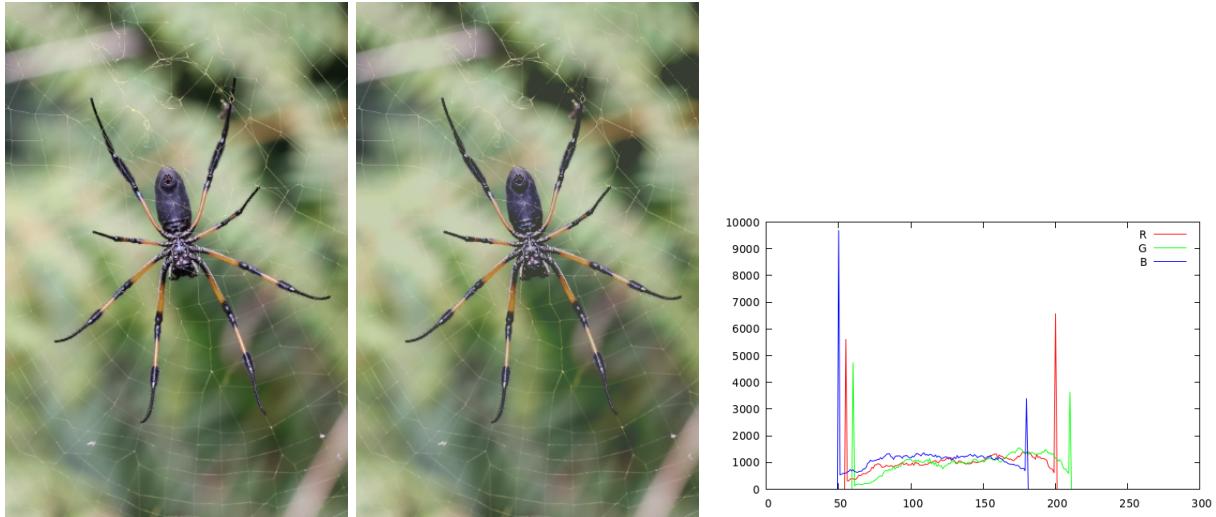


FIGURE 2.2 – Image originale -> Image extrema_seuilled -> Histogramme extrema_seuilled

L'image seuillée est légèrement plus terne que l'originale, c'est normal parce que les couleurs fortement sombre ou claire sont absentes de part les seuils utilisés.

On voit aussi très bien que les valeurs hors seuil s'écrasent sur les seuils.

Cela nous permet donc d'appliquer une expansion d'histogramme :

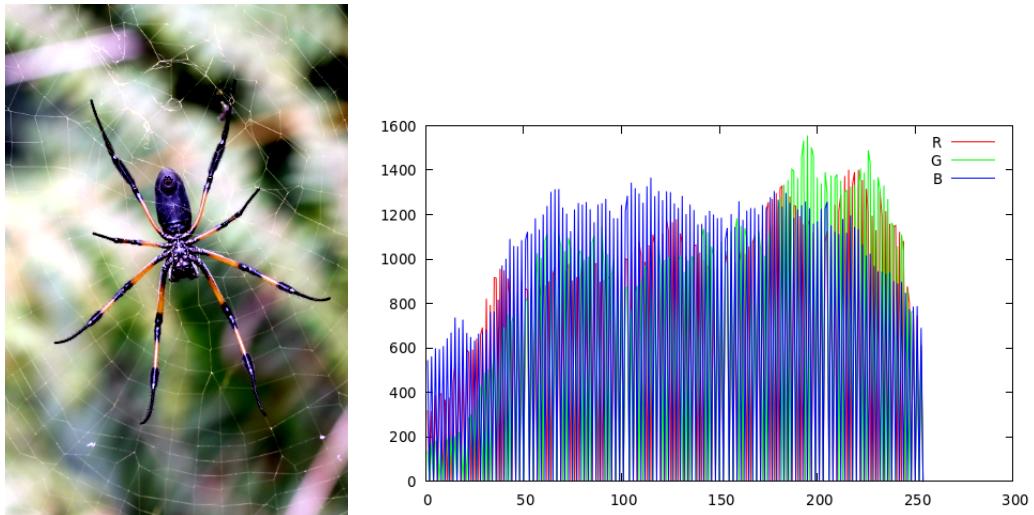


FIGURE 2.3 – Histogramme RGB de l'araignée

L'image finale est plus contrasté et coloré que les deux autres parce qu'on utilise l'expansion pour étendre l'amas d'intensité de couleur entre les seuils sur toute la plage des 0 à 255 pixels.

3. Egalisation des histogramme

Pour rappel, une égalisation d'image revient à ajouter du contraste dans une image.

En effet, l'objectif est d'avoir un histogramme le plus plat possible, ce qui aura pour conséquence d'homogénéiser les intensités de couleur.

Et par définition, une image contrasté contient le plus de nuances (et d'occurrence de nuance) de couleur possible. Donc pour arriver à égaliser notre histogramme, on décide d'utiliser sa fonction



FIGURE 3.1 – source : lcd-compare.com

de répartition.

On utilise la croissance de la fonction de répartition (qui est plus ou moins linéaire de 0.0 à 1) pour pondérer les valeurs initiale de l'image.

Cela aura pour effet de légèrement "calquer" ou "mapper" l'histogramme de l'image initiale sur la fonction de répartition, rendant les croissances et les décroissances plus douce.

Pour se faire on va simplement déterminer notre nouveau pixel p' à partir de p comme cela :

$$p' = FonctionDeRepartition[p] * 255$$

Voici nos résultats pour l'image de l'arbre :



FIGURE 3.2 – Arbre base(1er) et arbre égalisé(2eme)

On peut remarquer que les feuillages sont plus variant en niveau de gris ce qui donne effectivement du contraste à celui-ci mais l'image semble ne pas grandement faire ressortir l'effet de l'égalisation. Voici les histogrammes associés à chaque image :

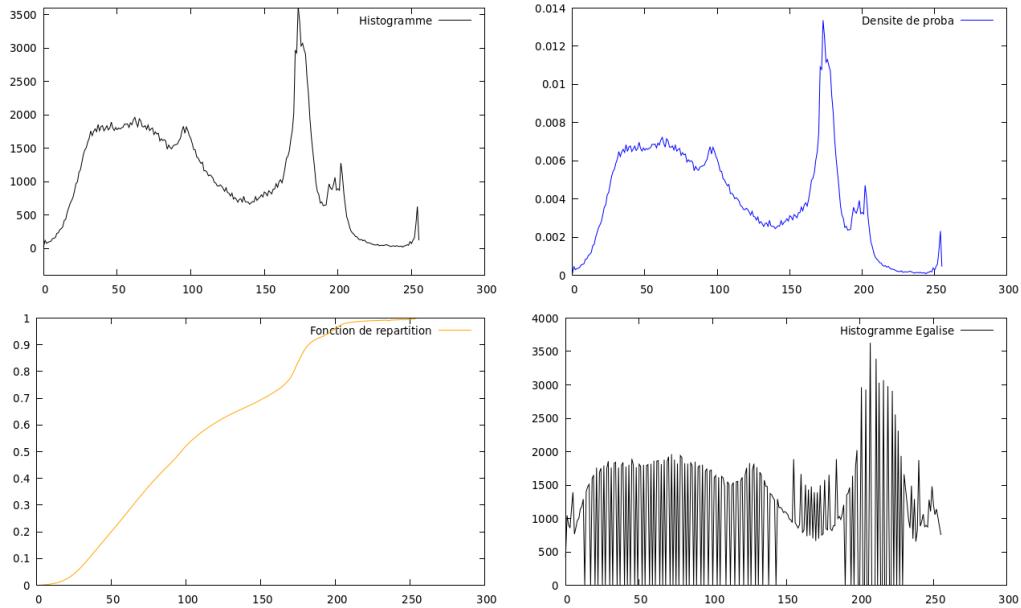


FIGURE 3.3 – Respectivement Histogramme de base -> densité de proba-> Fonction de repartition ->Histogramme de l'image égalisé

Un exemple plus parlant visuellement en terme de contraste est celui de lena :



FIGURE 3.4 – lena original et lena égalisée

Ce qui est éclairé semble l'être de manière plus intense et ce qui est dans le noir semble l'être aussi de manière plus intense.

4. Spécification

Maintenant, on s'intéresse à comparer deux images(1 et 2).

Pour se faire dans des conditions optimales, il faut que les deux images soit éclairés de manière identique afin d'avoir sensiblement le même spectre de niveau de gris, mais ce n'est pas toujours le cas.

Pour arranger cela, on décide d'utiliser l'image 1 comme image référence et l'image 2 comme image à étudier. On se base sur le principe suivant :

Image1 —Egalized1—> Image1Egalized1 —ReverseEgalized1—>Image1(malgré quelques variations dû aux arrondis)

Image2 —Egalized2—> Image2RefEgalized2 —ReverseEgalized1—>Image2 dans le spectre de greylevel de Image1

Le but étant d'utiliser la fonction inverse de l'égalisation pour "transposer" le spectre de niveau de gris de l'image1 sur l'image2 qu'on étudie afin d'avoir des choses comparable et traitable.

Pour l'égalisation, voir le chapitre précédent.

Pour la partie inversion de l'égalisation, on construit la fonction de répartition de l'image1 (la référence) et on détermine le résultat de ReverseEgalized1 comme cela :

```
1   for (int i = 0; i < nTaille; i++) // Pour tout les pixels de Img2
2   {
3       int p_ = Img2[i]; // On récupère le pixel de l'image2
4
5       //On retrouve la valeur inverse de la fonction de répartition
6       for (int j = 0; j < 256; j++) //Pour tout les niveau de gris possible
7       {
8           if (FdR_ref[j] >= ((float)p_ / 255)) // Dès qu'on dépasse le FdR_rer
9           {
10               reverse_FdR_Reverse = j; // On choppe le pixel correspondant as nouvelle intensité
11               break;
12           }
13       }
14       ImgOut[i] = reverse_FdR_Reverse;
15   }
```

Passons à la pratique avec image 1 lena.pgm et image2 tortue.pgm :

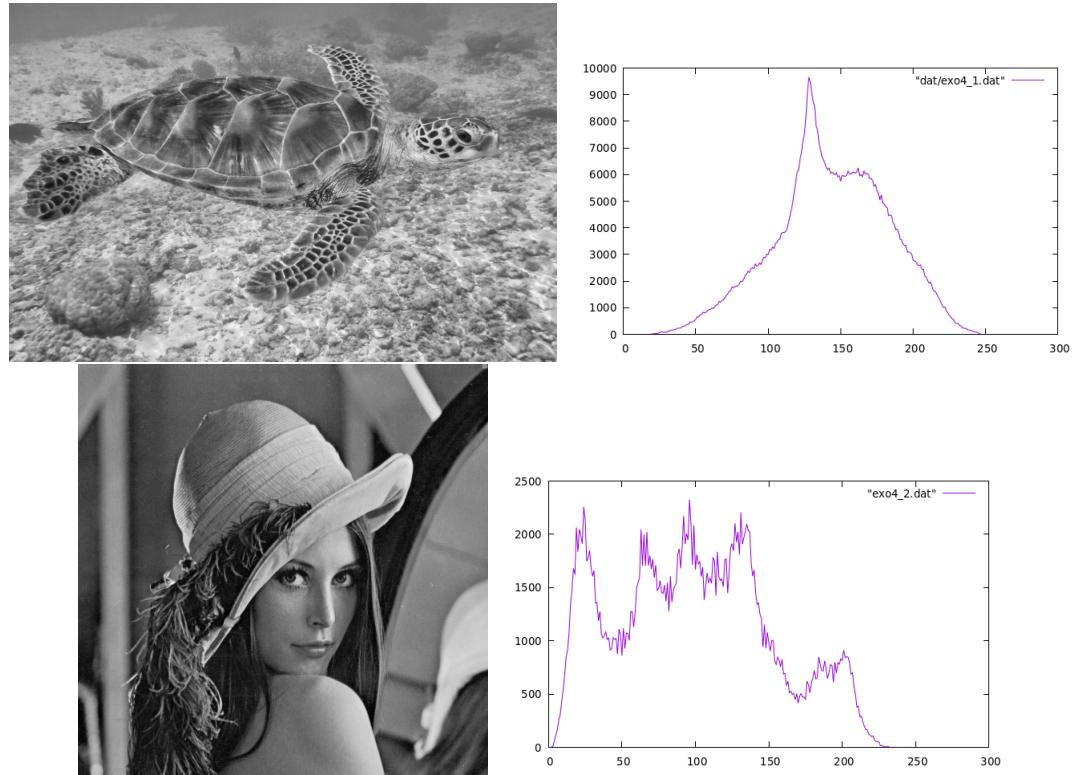


FIGURE 4.1 – Pas du tout le même spectre de niveau de gris, pas le même nombre de mod etc..

On égalise donc notre image de tortue :

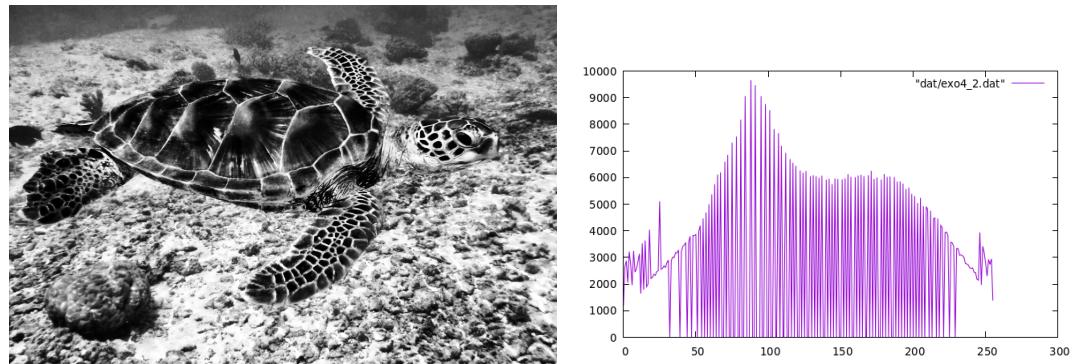


FIGURE 4.2 – L'histogramme et l'image de tortue égalisée par sa propre fonction de répartition

Enfin, on utilise la fonction inverse de répartition de lena pour spécialiser notre image :

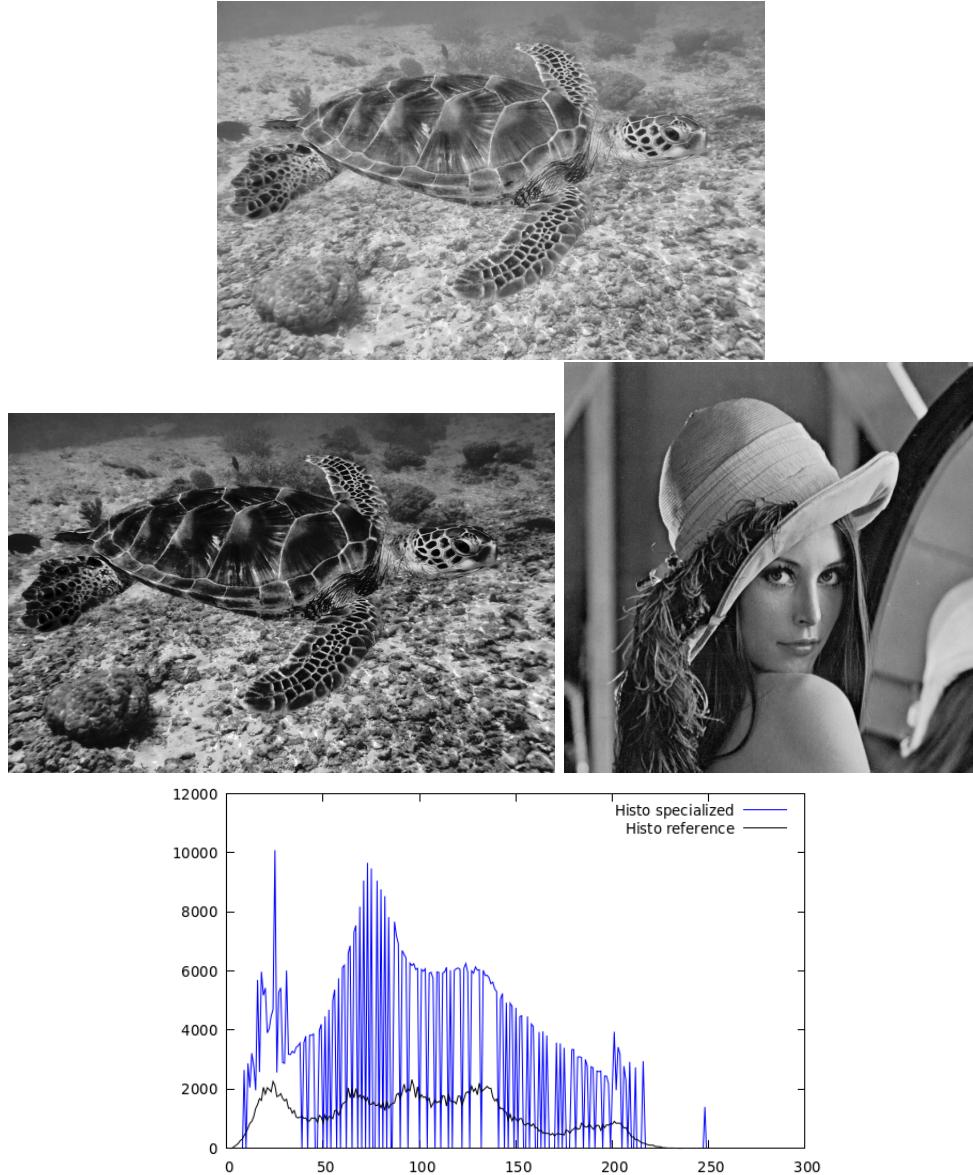


FIGURE 4.3 – L'image tortue de base, l'image tortue spécifié par lena et l'image de référence de lena + les histogrammes superposés des deux dernières images

On voit clairement avec l'ancienne image de tortue.pgm sous les yeux que l'image semble être bien s'être adapté au spectre des intensité de gris de l'image de lena. Et on peut le confirmer lorsque l'on regarde les histogrammes superposés avec celui de la tortue spécifiée (en bleu) qui épouse bien celui de lena.

On essaye un dernier exemple sur cette méthode de spécification avec une image beaucoup plus sombre et en plus une image ayant subit une expansion pour voir comment l'image spécifiée s'adapte :

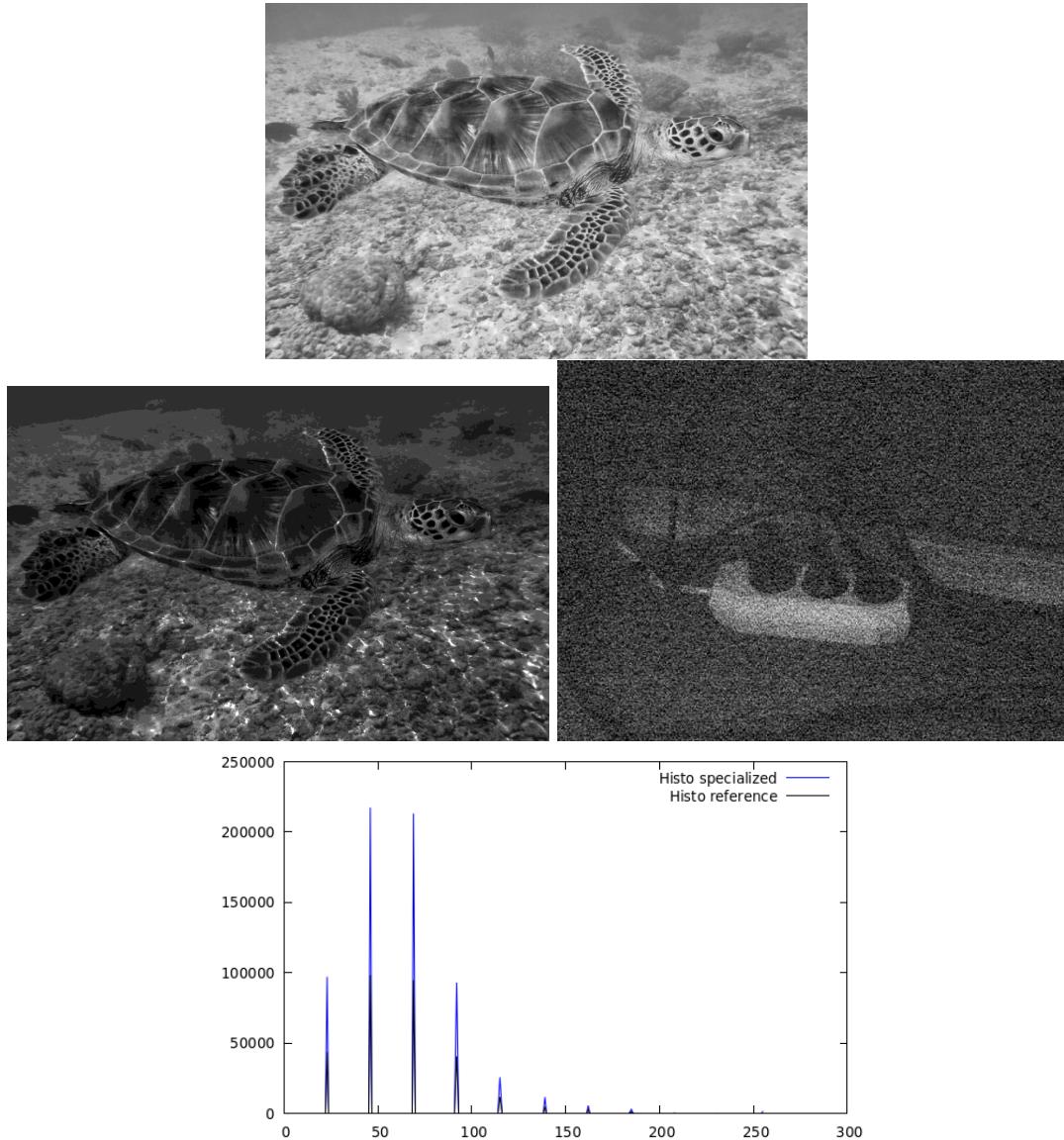


FIGURE 4.4 – L'image tortue de base, l'image tortue spécifié par blackexpanded.pgm et l'image de référence de blackexpanded + les histogrammes superposés des deux dernières images

Encore une fois, ça a l'air cohérent.

5. Conclusion

Dans ce tp, nous avons vu comment manipuler les histogrammes pour effecter des pré-traitement permettant de faciliter la visibilité ou la manipulation des intensités de couleur d'image ppm et pgm.