

Rapport TpG

Oxana USHAKOVA

April 10, 2017

1 Couleur

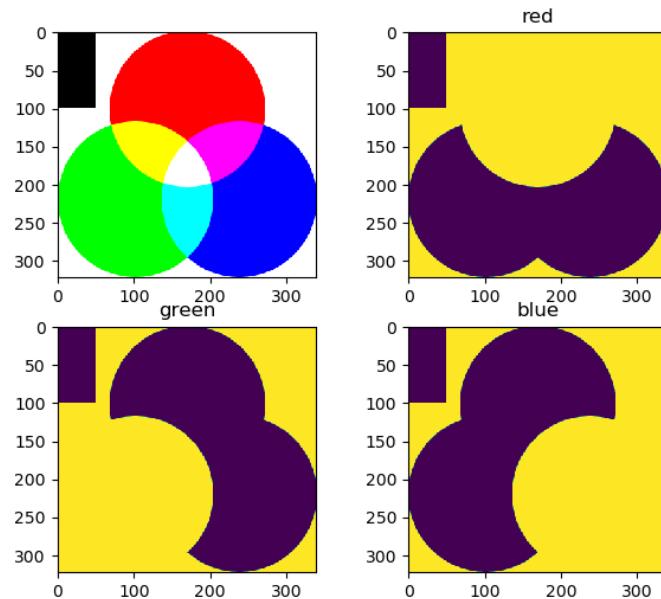
1.1 Schéma RGB

RGB est un des systèmes de codage des couleurs en informatique. Le mélange des couleurs primaires : red, green et blue - peut donner tout couleur possible, qui est représenté sous la forme d'un array de trois valeurs correspondantes qui varient de 0 à 255.

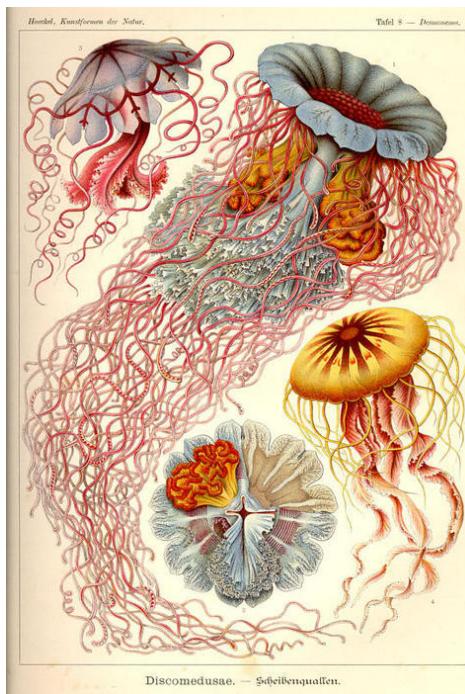
A l'aide du schema RGB on peut définir des nouveaux couleurs en Latex:

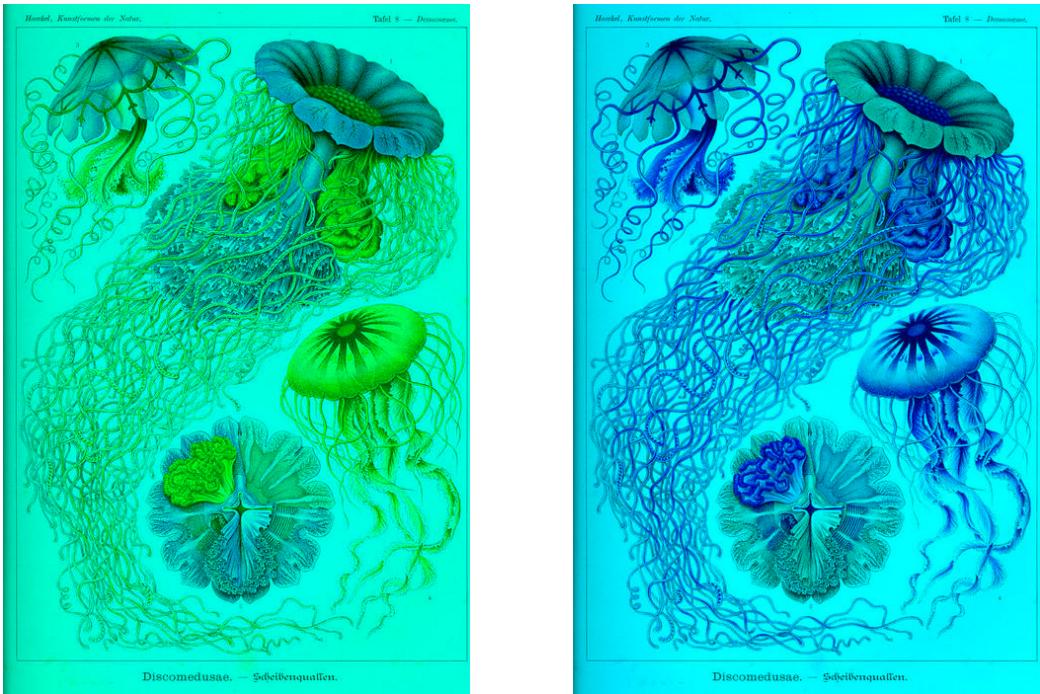
- 0,0,0 - Black
- 123,123,123 - Grey (moyennes des R,G et B)
- "White : 255,255,255
- **0,255,255 - No red**
- **255,0,255 - No green**
- **255,255,0 - No blue**

Regardons sur le schema RGB et décomposons-le en canaux, càd on affiche que les pixels où valeur d'un canal n'est pas nulle.

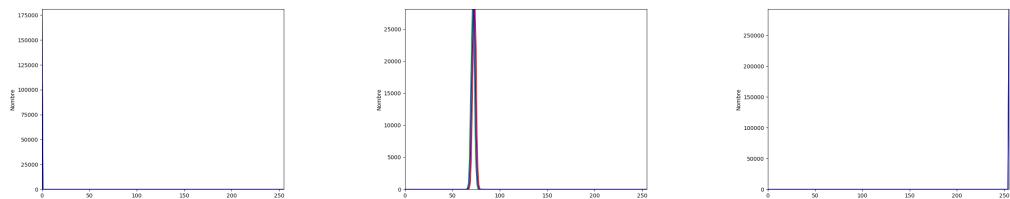


Prenons une autre image:

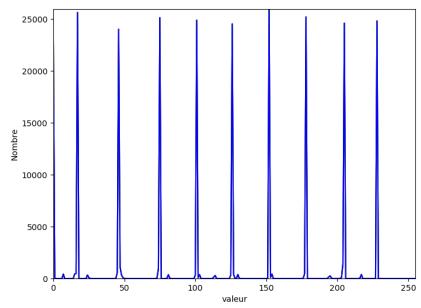
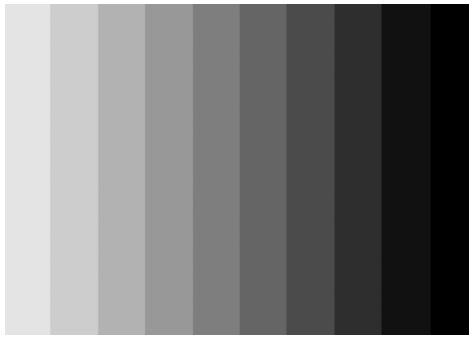
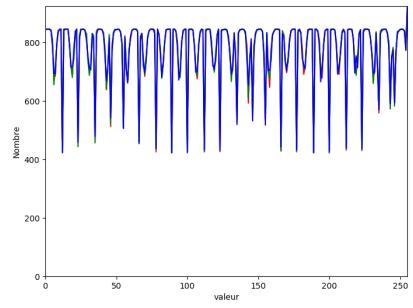




Utilisons un autre outil d'analyse - par histogrammes: elles représentent le nombre des valeurs des pixels du chaque couleur, canal.
D'abord, on regarde sur les histogrammes pour 0,0,0 et 75,75,75 et 255,255,255.



On reçoit le résultat attendu, tous les couleurs de gray-scale ont les composantes égales. Il est aussi intéressant des voir les histogrammes de gray-scale gradients: lisse et à l'escalier:



1.2 Quantization

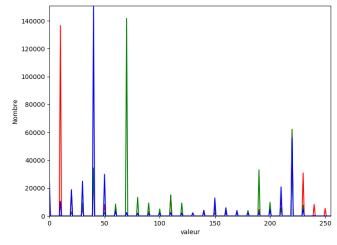
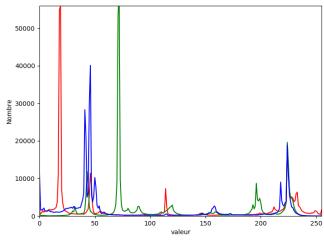
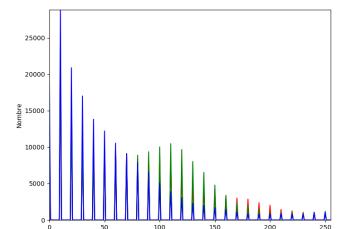
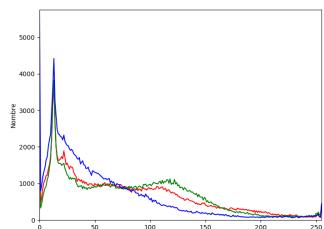
Quantization - est un moyen de compression d'une image où on réduit le nombre des couleurs distinctes. Si le taux de quantization est trop élevé la perte de qualité d'une image ne sera plus négligeable.

On peut implémenter cette technique en python avec la division sans reste (deux slashes):

```
ushakova@ushakova-HP-Notebook: ~
ushakova@ushakova-HP-Notebook:~$ python3
Python 3.4.3 (default, Nov 17 2016, 01:08:31)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 178/10
17.8
>>> 178//10
17
>>> 17*10
170
>>> 
```

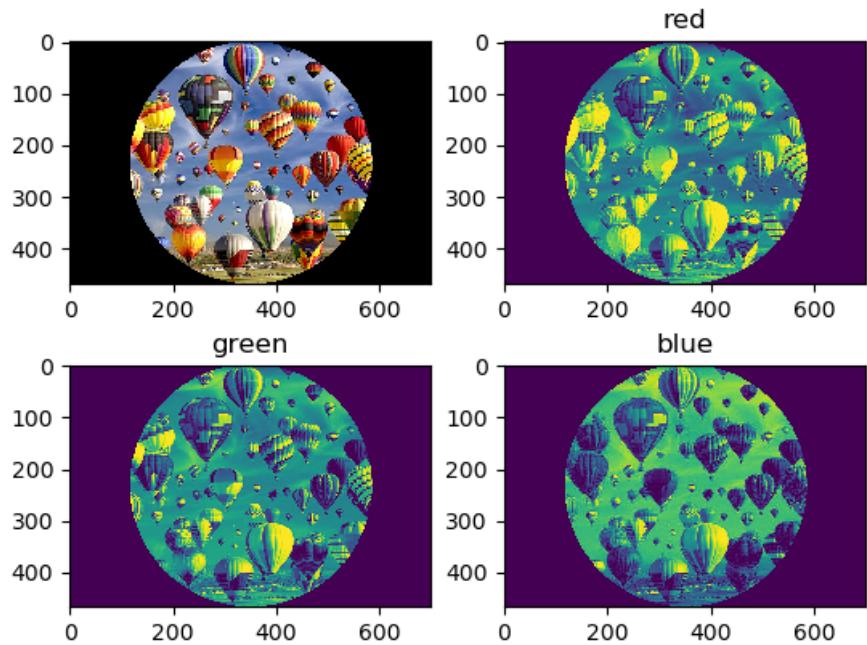
Prenons deux images: "hippie" qui a beaucoup de nuances de couleurs et "south park" dont la palette des couleurs est assez pauvre. Donc, la

quantization sera plus intéressante pour la première image -observons les histogrammes avant et après la compression:



1.3 Encerclement

L'encerclement d'une image et son représentation par canaux.



2 HSL, HSV

HSV (Hue Saturation Value) est un autre système de gestion des couleurs en informatique qui est basé sur la perception des couleurs.

Dans notre exemple on a la même image mais à saturations différentes. Regardons sur les Voici quelques examples des images décomposées par ces canaux:



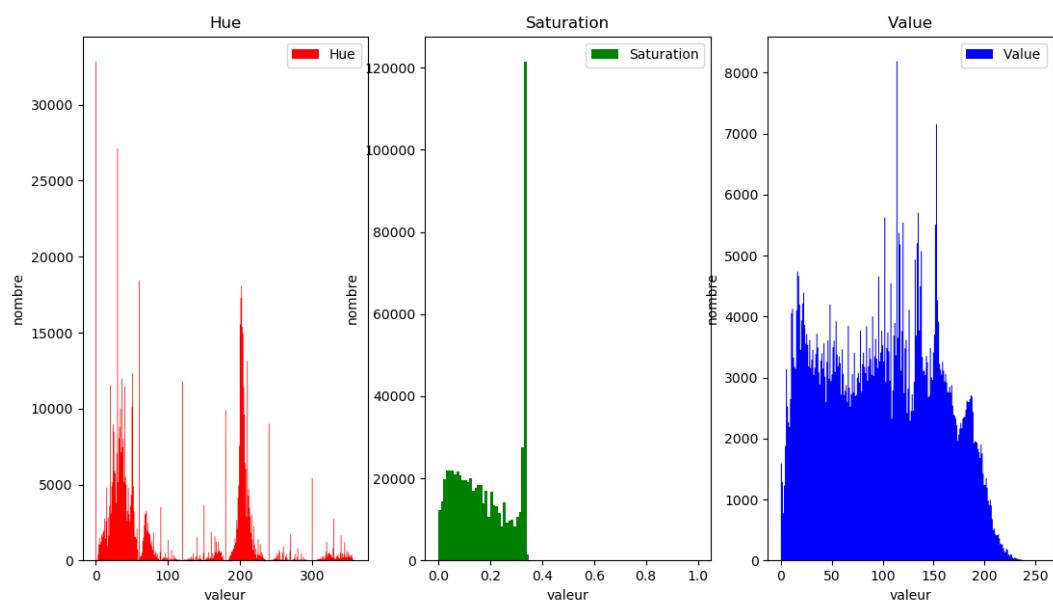
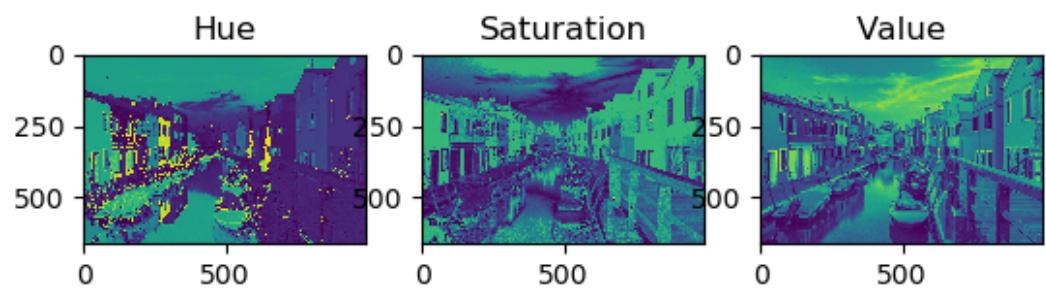


Figure 1: Low Saturation

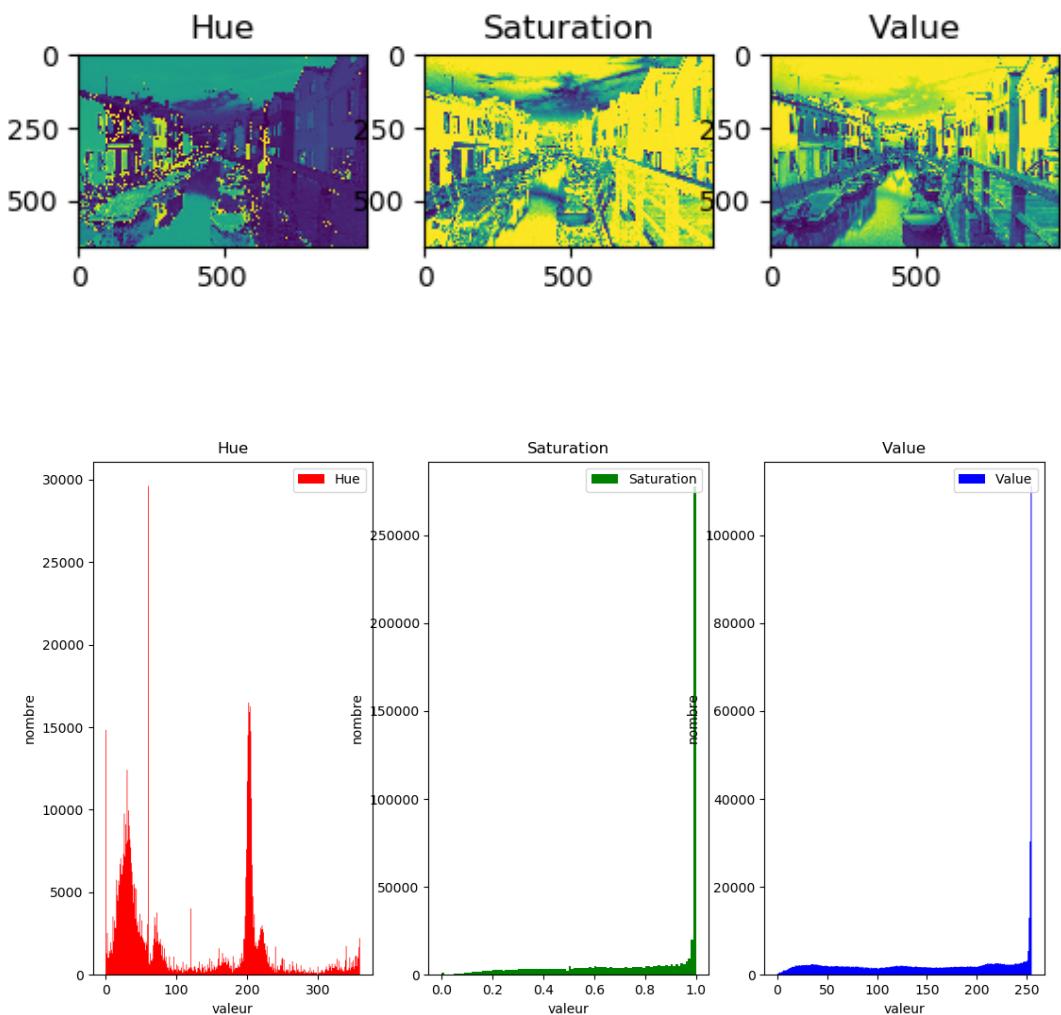
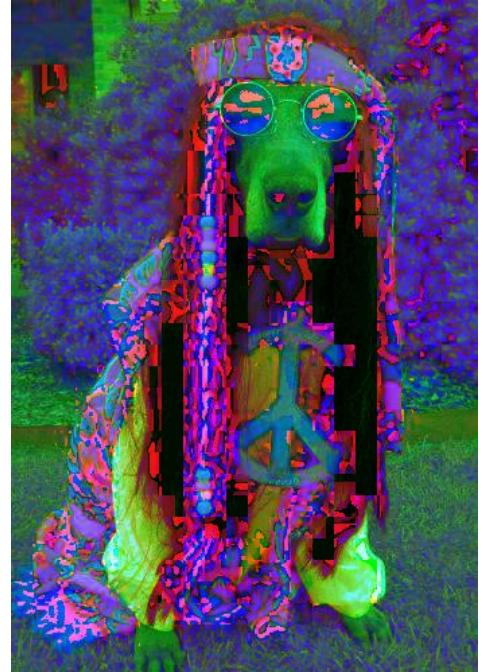


Figure 2: High Saturation



3 YCbCr

3.1 Perception

YCbCr est une manière de représenter l'espace colorimétrique en vidéo. On envoie Y, le signal de luminance (noir et blanc), plus deux informations de chrominance, Cb (Y moins le bleu) et Cr (Y moins le rouge). Le récepteur peut recréer le vert et reproduire une image couleur. En effet, si on a Y (rouge + vert + bleu) et Cb (Y - bleu) et Cr (Y - rouge), on peut mathématiquement recréer le vert en utilisant l'équation : $Y' = 0,3 R' + 0,6 G' + 0,1 B'$.

Pour transformer une image couleur en gray-scale il faut effectuer une moyenne pondérée des composantes RGB de l'image couleur. Les poids doivent prendre en compte le fait que l'oeil humain voit certaine couleur plus claire que d'autre. Il faut avoir un oeil d'expert pour différencier les poids naïfs et les poids officiels, mais on peut l'observer si on prend un schéma RGB:

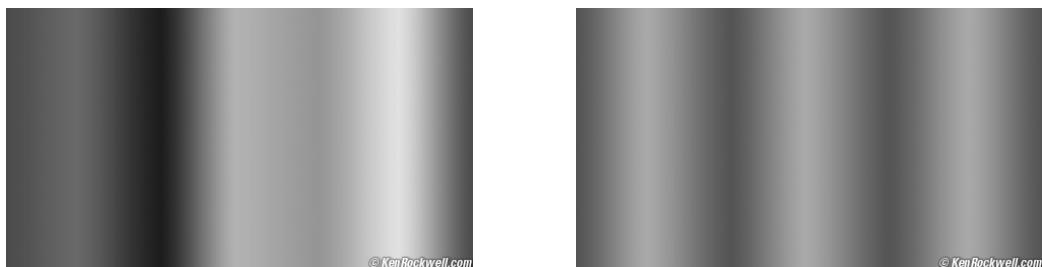
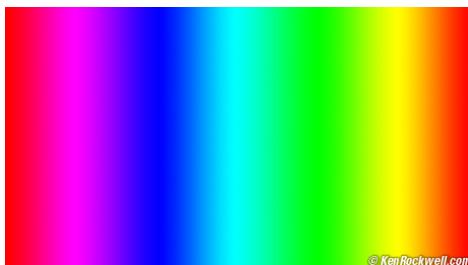
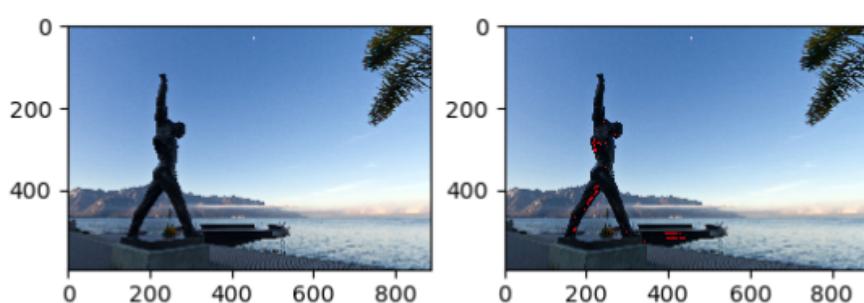
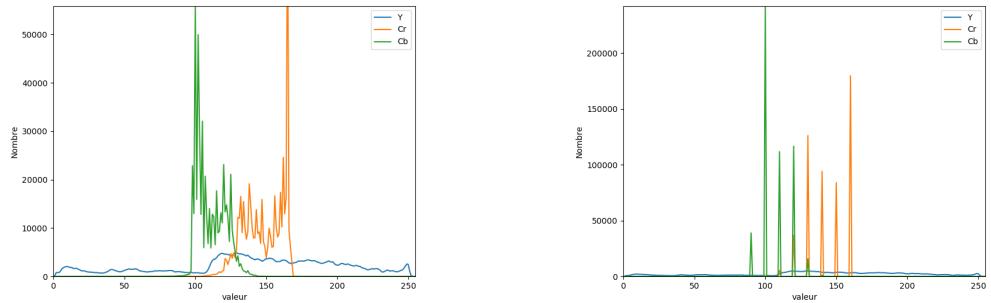


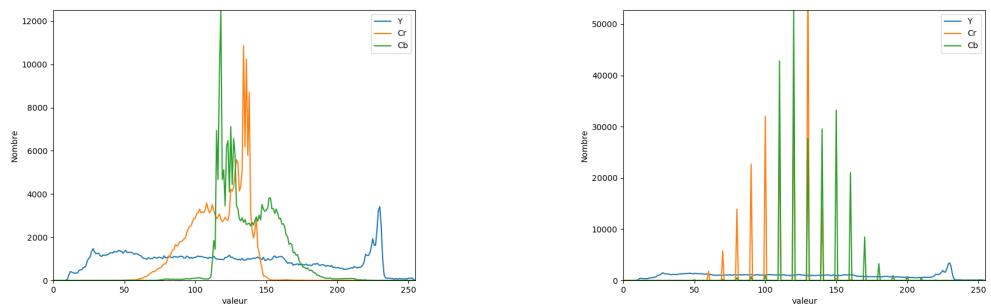
Figure 3: Noir et Blanc : officiels et naïfs

Comme la vision humaine est plus sensible à la luminance que la chrominance, la compression supplémentaire peut être obtenue en travaillant avec systèmes non-RGB comme YCbCr et faire la quantification sur les canaux (couleur et non-couleur). Maintenant, on applique la quantification aux composantes Cb et Cr, voici les histogrammes:





Remarque : j'ai essayé plusieurs formules de conversion, mais je reçois toujours l'image érronée, même sans quantisation.



3.2 Mirroring et Inversement des couleurs

