

Compression d'image via la méthode Run-Length-Encoding

Chaolei CAI

Paris VIII

April 13, 2020

1 Présentation de l'algorithme

2 Version améliorée de l'algorithme

3 Différence entre le RGB et le HSV

4 Performances

Qu'est qu'une image?

```
Struct Image {  
    unsigned long sizeX  
    unsigned long sizeY  
    Glubyte*data  
}
```



Compression?

200	200	42	35
-----	-----	----	----



2	200	1	42	1	35
---	-----	---	----	---	----

Glubyte buffer reçoit la première case

GLbyte compteur reçoit 0

Pour toutes les cases lues :

Si le buffer ne change pas:

Si le compteur n'atteint pas la limite :

Incrémenter le compteur,
passer à la case suivante

Sinon:

stocker le compteur et le buffer
reset le compteur à 1

Sinon :

stocker le compteur et le buffer
reset le compteur à 1
buffer reçoit la nouvelle valeur lue

Fin de la boucle

stocker le compteur et le buffer

Cas idéal



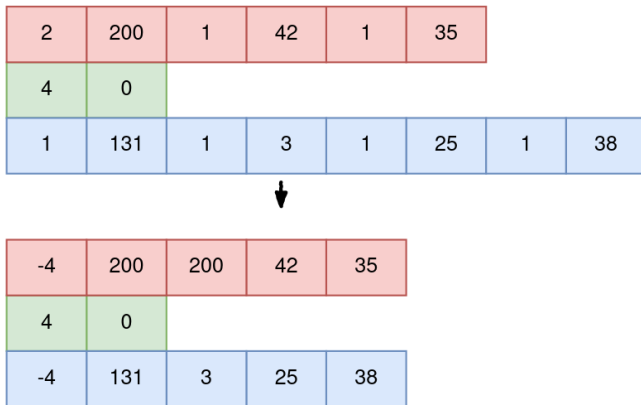
Pire cas possible

131	3	25	38
-----	---	----	----

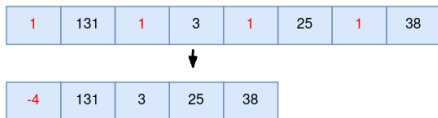


1	131	1	3	1	25	1	38
---	-----	---	---	---	----	---	----

Méthode améliorée par Silicon Graphics International



Seconde compression à appliqué sur le résultat précédente



Déclarer le tableau resultat

ptr_indice et ptr_vide reçoivent 0

Pour toutes les case d'indice :

Si l'indice est grande → stocker

ptr_indice reçoit ptr_vide

ptr_vide s'incrémente de 1

resultat [ptr_indice] reçoit l'indice lue

resultat [ptr_vide] reçoit la valeur de la case qui suit l'indice

ptr_vide s'incrémente de 1

Pseudo-code de la réduction

Déclarer le tableau resultat

ptr_indice et ptr_vide reçoivent 0

Pour toutes les case d'indice :

Si l'indice lue est grande → stocker

Sinon si resultat [ptr_indice] < 0 // autrement dit l'indice lue est petite et qu'il y a déjà une réduction en cours

ET l'ajout de l'indice ne fait pas déborder resultat[ptr_indice] :

→ Soustraire à resultat[ptr_indice] l'indice lue
Ajouter dans résultat autant de fois qu'il y a de répétition lue

Sinon : // il faut faire une réduction

ptr_indice reçoit ptr_vide

ptr_vide s'incrémente de 1

resultat [ptr_indice] reçoit l'opposé de indice lue

resultat [ptr_vide] reçoit la valeur de la case qui suit l'indice

ptr_vide s'incrémente de 1

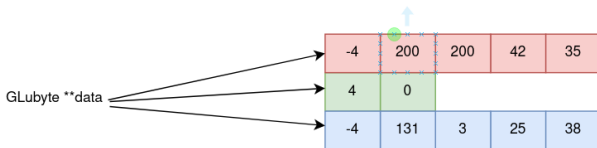
Ajouter dans résultat autant de fois qu'il y a de répétition lue

Fin de la boucle

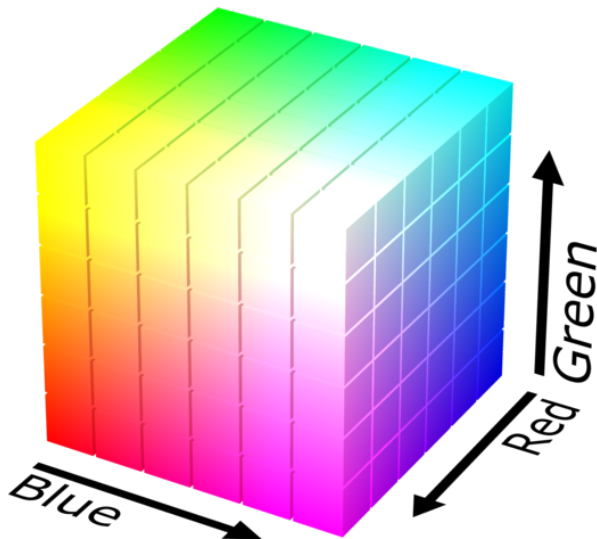
Nouvelle structure d'image

Struct Image {
unsigned long sizeX
unsigned long sizeY
Glubyte*data}

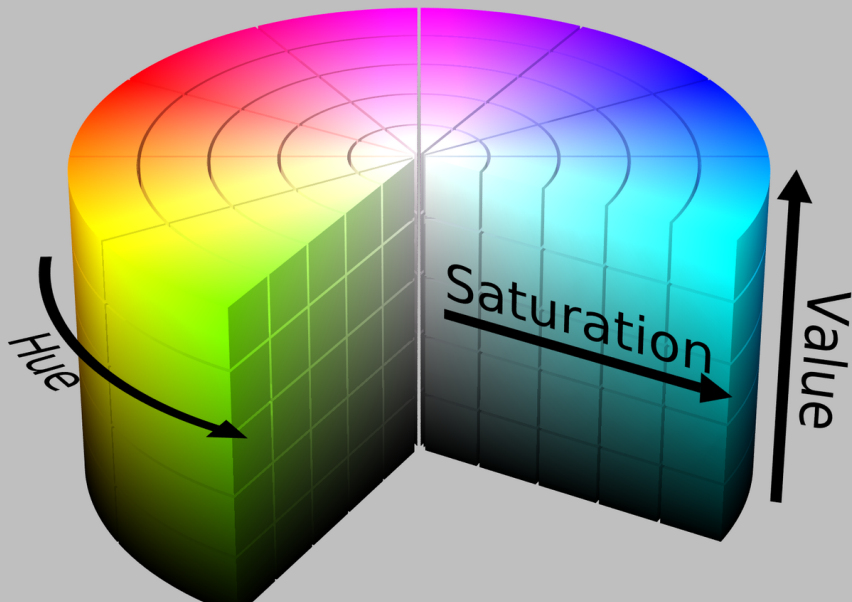
Struct Image_RGB_Compressed
unsigned long sizeX
unsigned long sizeY
unsigned long * ChannelSize
Glubyte ** data }



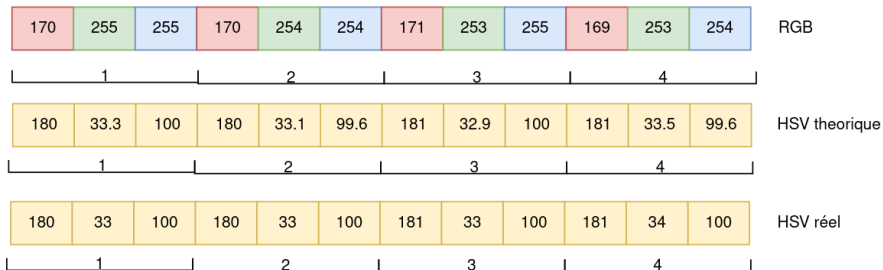
Cube RGB



Cylindre HSV



Avantages du hsv



Nom	Taille initial	RGB	HSV
can_bottom2	100%	95.32%	113.76%
chatou	100%	100.17%	131.99%
Cordiliere2_V3	100%	97.72%	115.95%
Kili_mais	100%	98.09%	125.51%
Refuges	100%	95.32%	117.35%
requin_leopard	100%	100.16%	131.63%
comic	100%	72.52%	88.15%
mickey	100%	26.11%	26.18%
morty	100%	49.76%	59.25%
morty_with_imagesave	100%	49.75%	59.25%
green	100%	1.57%	1.05%

Moyennes élaguées:

76% sous RGB, et **93%** sous HSV

Sources

Cube RGB

Cylindre HSV