



Département de génie informatique et de génie logiciel

**INF8770**  
**Technologies multimédias**

Travail pratique 2

Hiver 2020  
Section 01

Soukaina Moussaoui - 1728325  
Carine Thouin - 1837383

Le 31 mars 2020

## Question 1

L'effet positif du sous-échantillonnage 4:2:0 est que ce dernier offre un très bon taux de compression. En effet, étant donné que celui-ci ne gère qu'un seul échantillon parmi quatre, les données à gérer sont moindre, d'où la bonne performance de ce type de sous-échantillonnage. D'autre part, parmi ses points négatifs, nous retrouvons la création d'artéfacts qui donnent un aspect dégradé de l'image originale. Ces artéfacts sont dûs à la perte d'échantillons de chrominance.

À l'inverse, parmi les effets positifs du sous-échantillonnage 4:4:4, ce dernier n'a aucune perte de données, donc les artéfacts de l'image originale sont moindre par rapport au sous-échantillonnage 4:2:0. En effet, l'échantillonnage s'effectue sur chaque pixel de l'image. La qualité de cette dernière est alors conservée. Néanmoins, étant donné qu'on a plus de données à gérer, le taux de compression sera moindre par rapport au 4:2:0.

Un changement de l'espace de couleur avant un sous-échantillonnage permet de diminuer la taille des composantes de chrominance d'une image. La raison principale de cette diminution est que l'oeil humain perçoit plus facilement la luminance que la chrominance. De ce fait, en réduisant le nombre de composants de l'image et en effectuant un sous-échantillonnage, nous pouvons obtenir une bonne qualité d'image.

## Question 2

Le but principal de la DCT est d'exprimer les composantes corrélées d'une image dans une base où elles peuvent être plus ou moins indépendantes l'une de l'autre. Dans le contexte de JPEG, l'utilité principale de la DCT est de pouvoir discerner les données importantes des moins importantes. Ces dernières pourraient alors être effacées par la suite afin de gagner de l'espace et la qualité de l'image serait conservée du point de vue de l'oeil humain.

Il est important de savoir que lorsqu'on effectue une DCT, les informations importantes sont souvent concentrées dans les basses fréquences plutôt que dans les hautes fréquences. Il est alors possible de représenter une même image en éliminant les données de haute fréquence, car l'oeil humain est moins sensible à ces dernières.

Ainsi, la DCT permet de changer les pixels d'une image en fréquence et coefficients. Elle ne permet pas alors de compresser une image, mais simplement de la préparer à la prochaine étape qui est la quantification, dans laquelle les données de haute fréquence seront effacées.

## Question 3

### 3.1.)

Il est possible de voir des artefacts sur des images lorsque nous effectuons une compression et une décompression dessus. Cela peut être dû au fait que la compression et/ou la décompression ne s'est (sont) pas très bien faite(s). La cause principale de l'apparition de ces artefacts est la perte de données lors de la compression. Lors de la décompression, nous retrouvons alors des imperfections dans quelques endroits de l'image décompressée dans lesquels les données ont été perdues.

Comme nous l'avons vu à la première question, lors de la transformation de RGB à YCbCr, la perte de données peut avoir lieu si nous utilisons un sous-échantillonnage 4:2:0 par exemple. En effet, nous ne gèrons qu'un seul échantillon de chrominance sur 4 de l'image. On peut aussi avoir une perte de données lors de la DCT si les valeurs sont arrondies (par exemple: pour 0,235739 on arrondira à 0,24). Finalement, il y aura une perte de données lors de la quantification étant donné que celle-ci enlève toutes les hautes fréquences pour lesquelles l'oeil humain est moins sensible.

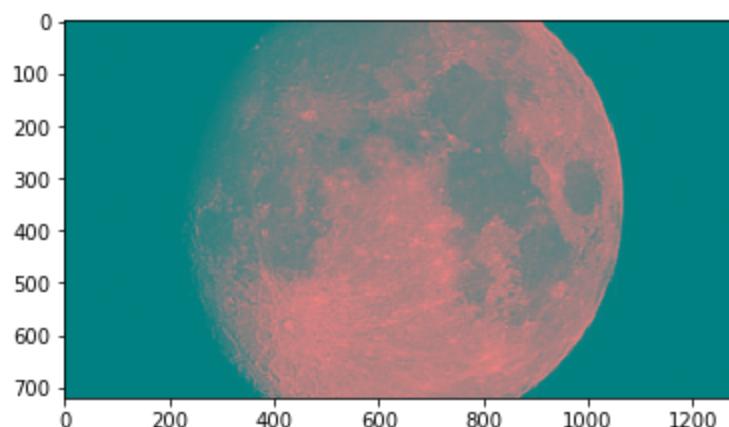


Figure 1: Image avec artefacts autour de la Lune avec la matrice 1

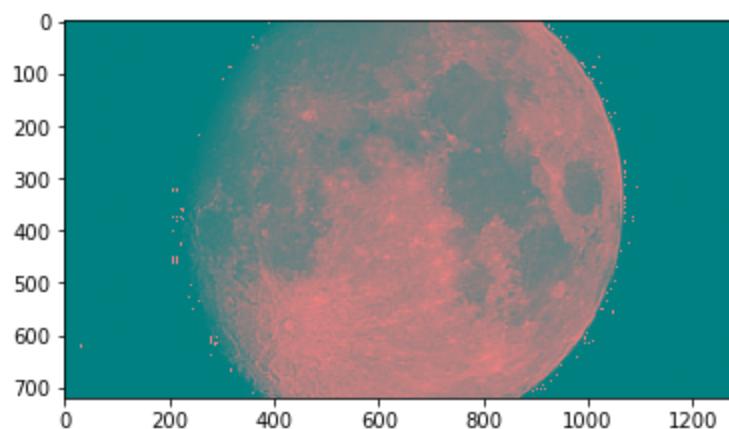


Figure 2: Image avec artefacts autour de la Lune avec la matrice 2

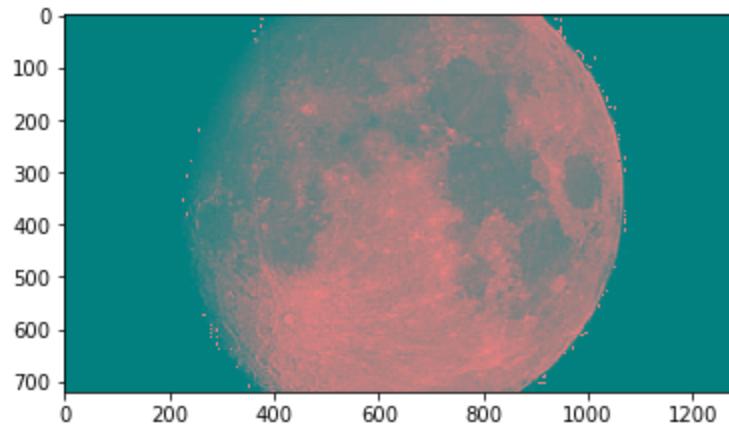


Figure 3: Image avec artéfacts autour de la Lune avec la matrice 3

### 3.2.)

#### Matrice 1

```
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1
```

#### Matrice 2

```
16 11 10 16 24 40 51 61  
12 12 14 19 26 58 60 55  
14 13 16 24 40 57 69 56  
14 17 22 29 51 87 80 62  
18 22 37 56 68 109 103 77  
24 35 55 64 81 104 103 92  
49 64 78 77 103 121 120 101  
72 92 95 98 112 100 103 99
```

**Matrice 3**

```
53 37 33 53 80 133 170 203  
40 40 47 63 87 193 200 183  
47 43 53 80 133 190 230 187  
47 57 73 97 170 290 267 207  
60 73 123 187 227 363 343 257  
80 117 183 213 270 347 377 307  
163 213 260 290 343 403 400 337  
240 307 317 327 373 333 343 300
```

Dans ce travail pratique, nous avons utilisé trois types de matrices de quantification. La première étant composée que de 1 n'a pas d'impact sur la qualité de l'image ni sur la compression de cette dernière. Il y aura alors une perte de données très négligeable. D'autre part, la matrice 2 générera plus de pertes de données, donc plus de zéros. La compression de l'image sera meilleure que la précédente. Par contre, la perte de données aura un impact sur la qualité de l'image. Finalement, la matrice 3 offre un taux de compression inférieur à celui de la matrice 2. Néanmoins, la qualité de l'image est moins bonne. La matrice 3 n'offre pas une compression avec un bon rapport qualité/compression.

Le tableau suivant présente des données utiles aux questions 3.2 et 4, en utilisant les différents types d'images demandés à la question 4.

*Tableau 1 : Taux de compression selon la matrice de quantification et l'image*

| Types d'images | Image | Matrice 1 | Matrice 2 | Matrice 3 |
|----------------|-------|-----------|-----------|-----------|
| Standard       |       | 0.2593    | 0.3883    | 0.3839    |
| Très colorée   |       | 0.3913    | 0.4934    | 0.4587    |
| Dégradée       |       | 0.4971    | 0.5744    | 0.5365    |
| Noire          |       | 0.7656    | 0.7656    | 0.7656    |

## Question 4

Afin d'évaluer l'efficacité de notre algorithme de compression, nous devons tenir compte de différents paramètres : les différentes couleurs qui la composent, la manière dont elles sont distribuées à travers l'image et leur intensité. Si nous prenons l'exemple de l'image du paysage, nous pouvons remarquer que celle-ci possède un grand éventail d'intensités de couleurs et leur distribution est très variée. Si nous effectuons une faible quantification (matrice 1) sur une image de ce type, nous possédons un très faible niveau de compression, soit de 0.2593. Nous possédons un taux de compression un peu plus acceptable seulement si nous utilisons une matrice de quantification lourde telle que la matrice 2.

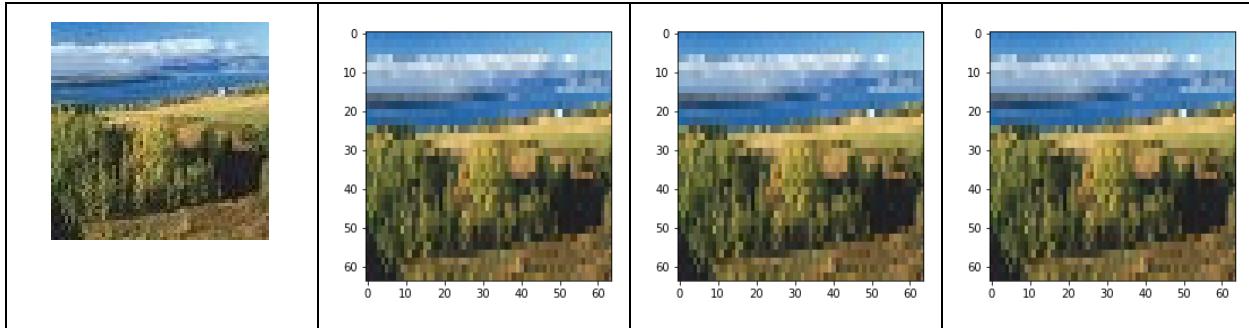
D'autre part, si nous prenons l'exemple de l'image noire, nous pouvons constater qu'elle possède le meilleur ratio de compression et ce, même avec une matrice de quantification composée que de 1. Nous pouvons expliquer cela par le fait que, étant donné qu'il n'y a qu'une seule couleur, et donc aucune variation ainsi qu'une distribution à 100% uniforme de la même couleur, la compression à l'aide des algorithmes de Huffman et RLE se fait de manière très efficace.

Finalement, nous pouvons remarquer que l'image choisie en dégradé et l'image colorée possèdent aussi un faible taux de compression si nous la quantifions avec la matrice 1. Ceci est dû au grand contraste entre les couleurs qui les rend difficilement quantifiables. Néanmoins, le taux de compression s'améliore si nous utilisons une matrice lourde comme la matrice 2. Nous pouvons passer d'un taux de compression de 0.4971 à 0.5744, ce qui est assez bon.

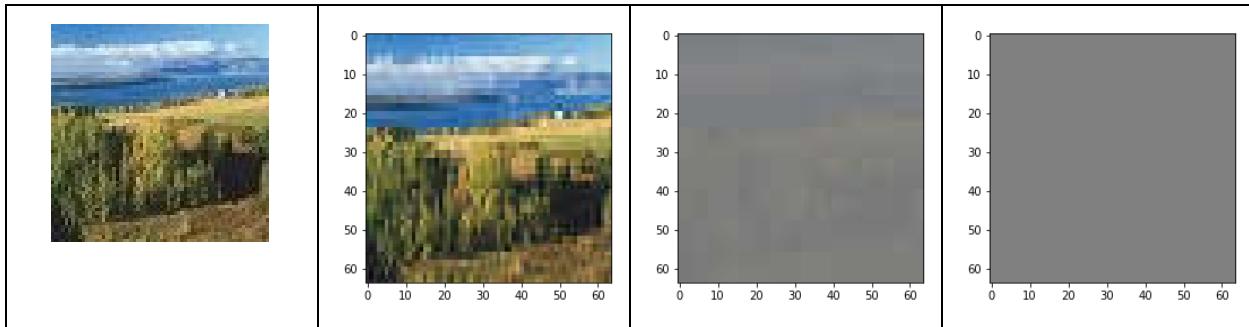
Avec toutes ces expériences, nous pouvons conclure qu'un grand nombre de couleurs dans une image ainsi qu'une distribution très hétérogène de ces dernières réduisent l'efficacité de la compression de l'image.

## Question 5

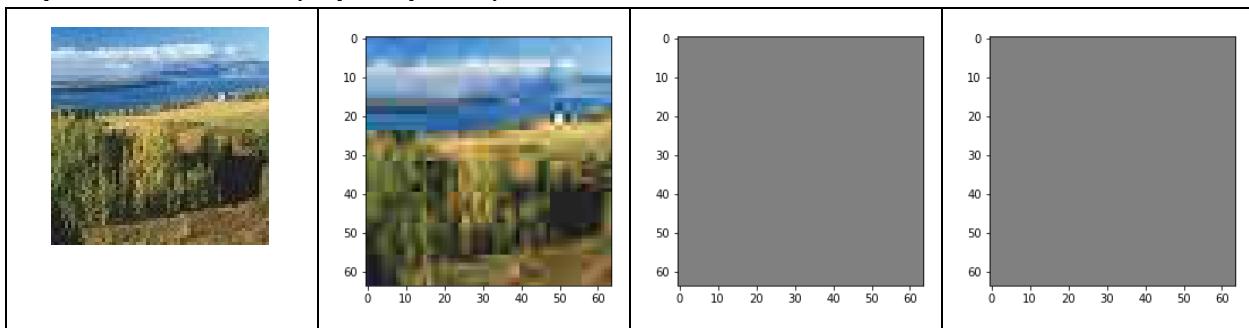
### Répétition matrice 1 (haute qualité)



### Répétition matrice 2 (basse qualité)



### Répétition matrice 3 (la pire qualité)



### Alternances de matrices

| Image normale | Matrice 2 | Matrice 1 | Matrice 3 | Matrice 1 |
|---------------|-----------|-----------|-----------|-----------|
|               |           |           |           |           |

Comme nous pouvons le remarquer, un grand nombre de quantifications sur une seule image provoque une grande perte de données ce qui impacte la qualité de l'image. Dans notre cas, la qualité de l'image dépend non seulement du nombre de répétitions de la quantification, mais aussi de la matrice utilisée. Comme nous pouvons le constater avec la matrice 1, celle-ci diminue la qualité de l'image, mais le nombre de répétitions ne dégrade pas encore plus sa qualité. Néanmoins, dans le cas de la matrice 2 et 3, la qualité de l'image est drastiquement diminuée quand nous la quantifions plus qu'une fois. Nous pouvons remarquer qu'à l'avant dernière quantification, avec la matrice 2 nous conservant une moins mauvaise qualité qu'avec la matrice 3. Ceci nous confirme que la matrice 3, même avec un taux de compression inférieur à celui de la matrice 2, réduit énormément la qualité de l'image (toute grise, nous ne pouvons même pas apercevoir légèrement le paysage). Finalement, en effectuant une alternance de quantifications avec la matrice 2 suivie de la matrice 1, nous remarquons une grande diminution de la qualité de l'image d'une itération à une autre et ce, même si la matrice 1 fait une très faible quantification. La quantification avec la matrice 3 vient empirer la qualité car, encore une fois, le paysage n'est même plus visible.

## Question 6

Afin de calculer la valeur du PSNR et SSIM pour JPEG2000 dans le code implémenté, nous avons dû convertir les images *.jp2* en *.jpeg* au préalable, car nous n'arrivions pas à lire les images *.jp2*. Nous avons utilisé le lien suivant à cet effet : <https://convertio.co/>.

Tableau II: Comparaison des valeurs PSNR et SSIM pour JPEG et JPEG2000

| Nom du fichier | Matrice | JPEG |       | JPEG 2000  |            |          |          |
|----------------|---------|------|-------|------------|------------|----------|----------|
|                |         | PSNR | SSIM  | small PSNR | small SSIM | big PSNR | big SSIM |
| kodim02        | 1       | 9.90 | 0.016 | 26.42      | 0.65       | 39.87    | 0.96     |
|                | 2       | 9.90 | 0.017 |            |            |          |          |
|                | 3       | 9.97 | 0.015 |            |            |          |          |

Pour cet exercice, nous avons choisi l'image *kodim02* et nous avons calculé sa qualité selon deux méthodes de compression, soient : JPEG et JPEG2000 ainsi qu'en fonction des trois matrices présentées dans les exercices précédents. Dans le cas de JPEG2000, nous avons effectué les calculs du PSNR et du SSIM selon la taille de l'image aussi.

Comme nous pouvons le constater dans le tableau, les valeurs du PSNR et du SSIM du pipeline codé (JPEG) ne varient quasiment pas en fonction des matrices. Cela veut dire que le choix des matrices, aussi différentes soient-elles, n'a pas un impact significatif sur la valeur des calculs de qualité d'images selon les méthodes PSNR et SSIM.

En outre, si nous comparons les valeurs du PSNR et SSIM dans le contexte de JPEG2000 et en fonction de la taille de l'image, nous remarquons que les grandes images ont des valeurs supérieures que celles des grandes images. Nous pouvons affirmer que les grandes images ont une meilleure qualité en termes de PSNR et SSIM que les petites.

Finalement, en comparant les valeurs du pipeline codé JPEG et celles de JPEG2000, nous constatons que JPEG2000 a de meilleurs indices PSNR et SSIM que JPEG et ce, quelque soit la matrice de quantification utilisée en JPEG. Nous pouvons ainsi affirmer que la qualité d'images en JPEG2000 est meilleure que celle de JPEG.

## Références

- <https://pypi.org/project/SSIM-PIL/>
- <https://stackoverflow.com/questions/2659312/how-do-i-convert-a-numpy-array-to-and-display-an-image>
- <https://dsp.stackexchange.com/questions/38065/peak-signal-to-noise-ratio-psnr-in-python-for-an-image>