



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE

1873653 – DALPHOND, Jean-Olivier
1928777 – SARWAT, Mariam

INF8770 - RAPPORT DE TRAVAIL PRATIQUE #2

Travail présenté à M. Hugues Perreault

École Polytechnique de Montréal
2 novembre 2020

Note : Pour rouler le code utiliser la ligne suivante : « ./Pipeline.py -i cheminDeLimage »

Question 1

Un sous-échantillonnage 4:2:0 permet, en général, de mieux compresser une image sans trop perdre de qualité. Plus précisément, ce type de sous-échantillonnage offre un quotient de compression de 2:1. En d'autres mots, la taille d'une image est diminuée de moitié. Toutefois, pour des images possédant un haut contraste entre les couleurs (par ex. du texte sur un fond d'une autre couleur), la compression devient très apparente. Cela est surtout vrai pour les couleurs rouge et bleu. Si on compare avec un sous-échantillonnage 4:2:2 (quotient 3:2), on réalise qu'un sous-échantillonnage 4:2:0 offre un taux de compression plus élevé. On obtient donc un gain de compression lors de l'utilisation de ce dernier.

Cependant, on peut voir que le contraste bleu/rouge est relativement comparable entre les deux; on n'obtient pas une bonne qualité pour ce cas de figure, dans ni un ni l'autre des sous-échantillonnages. On observe donc une corrélation entre le contraste et la qualité de la photo résultante. En effet, lorsque le contraste entre les couleurs des pixels collés est moins grand, l'impact sur la qualité de la photo résultante est minime. On sait que la vision humaine est plus sensible aux variations de luminosité qu'aux variations de couleur. Ce fait peut être utilisé afin de mieux compresser une image. Ceci est réalisé en consacrant une plus grande part des données à la luminance qu'à la chrominance et ainsi économiser en espace. Pour ce faire, on doit faire une conversion en Y'CbCr (YUV) afin de rendre indépendant les composantes de couleur. Ceci permet, par la suite, de pouvoir modifier plus spécifiquement la luminance, soit Y'. Pour illustrer ces constats, nous avons utilisé les exemples suivants du site web RTINGS.com (Babcock, 2019) :

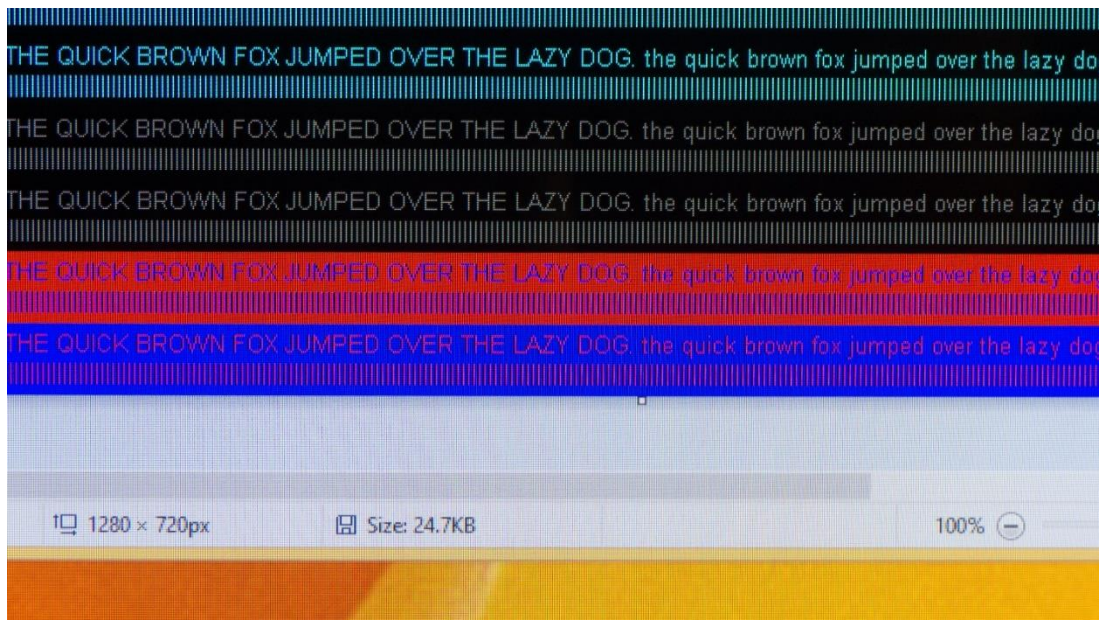


Figure 1 - Image originale 4:4:4

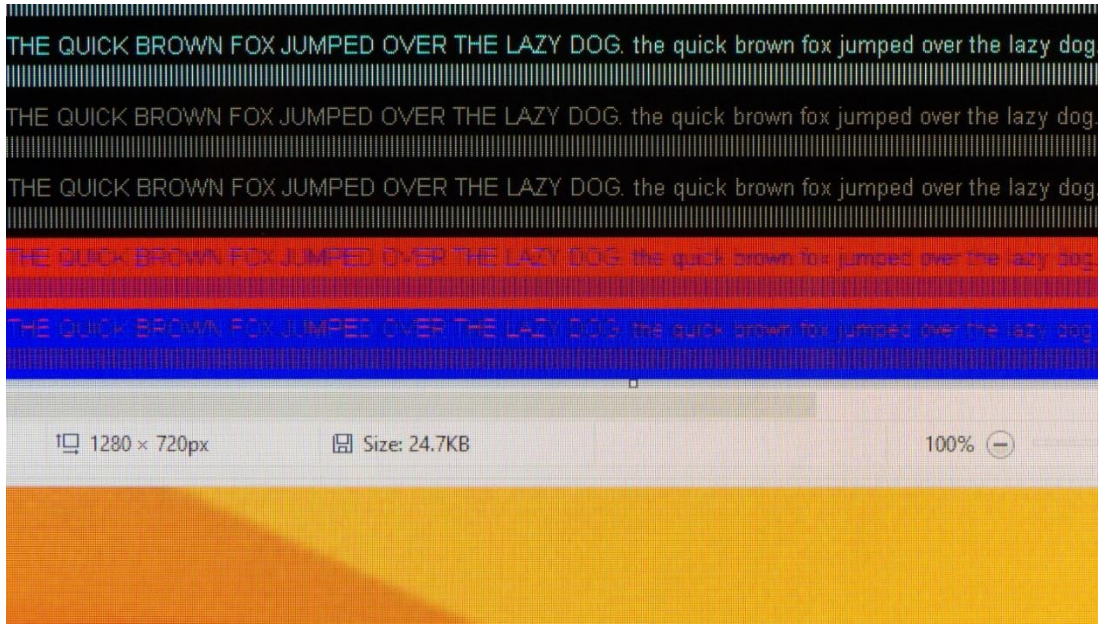


Figure 2 - Sous-échantillonnage 4:2:2

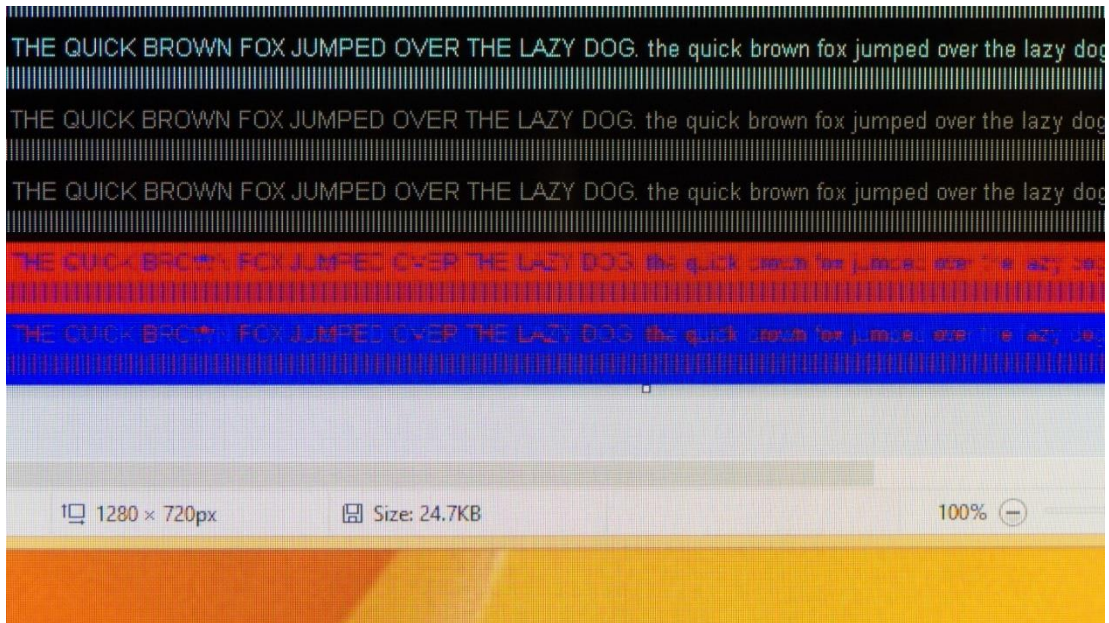


Figure 3 - Sous-échantillonnage 4:2:0

Question 2

Le fait d'utiliser une transformée en ondelettes discrètes (comme celle de Haar) permet d'obtenir une multitude de valeurs extrêmement basses (voire nulles) dans trois quadrants d'une image. En effet, on peut aller chercher plus de valeurs nulles en augmentant le n , soit le niveau de récursion (un $n = 0$ indique l'image originale).

La transformée en ondelettes ne compresse pas, en soi, l'image. Il s'agit d'une opération matricielle réversible, donc sans perte ou avec des pertes minimales lorsque la limite de précision du type numérique utilisé est atteinte. En pratique, on choisit souvent un algorithme avec un peu de pertes, mais ce n'est pas toujours le cas. En effet, cette transformée déplace l'information de l'image, et par les étapes subséquentes, cela permettra une compression plus efficace (moins de perte de qualité perceptible). Par ce processus, on concentre les valeurs plus significatives ensemble, et les valeurs moins significatives ensemble. Ensuite, afin d'éliminer le poids de ces valeurs moins significatives, on applique un quantificateur à zone morte. En fait, ce dernier permet d'arrondir ces valeurs à zéros lorsqu'ils se retrouvent dans sa zone (soit la « zone morte »).

Il est important de se rappeler que cette transformée ne compresse pas ou peu les données. Par conséquent, il ne suffit pas d'appliquer le DWT avec un niveau de récursion très élevé. Effectivement, cela peut entraîner des pertes au niveau du quantificateur. C'est pour cela que plus le n augmente, plus la taille de la zone morte doit diminuer (pour garder une qualité plus ou moins constante).

Question 3

Le niveau de récursion (n) pour un DWT de Haar n'affecte pas d'une façon significative le taux de compression d'une image lorsqu'on utilise un quantificateur qui implique peu de perte. Effectivement, ceci est grâce au fait que ce type de transformée est réversible. En d'autres mots, le niveau de récursion n'affecte pas la réversibilité.

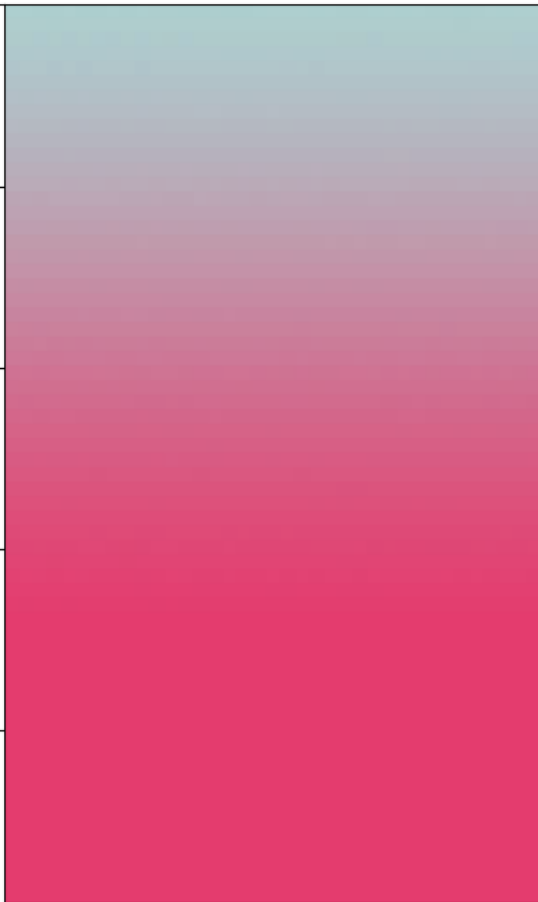
En effet, comme il a été mentionné, le DWT ne permet pas réellement de faire la compression d'une image. On n'aperçoit donc pas une différence entre une image transformée avec $n = 0$ et une transformée avec $n = 8$. La compression en tant que telle se fait à d'autres étapes, comme le codage (LZW par exemple). Plus précisément, la récursion permet d'accentuer ce regroupement de valeurs plus significatives et moins significatives. En pratique, on peut même amener la récursion jusqu'à traiter les pixels individuels (Flores-Mangas, 2018). Plus simplement, on veut diminuer l'entropie ce qui rend la compression à l'aide d'un codage par dictionnaire (LZW par exemple) plus efficace. De plus, plus le niveau de récursion est haut, plus le taux de compression augmente après l'application d'un codage tel quel.

Question 4

4.1) Les images qui sont présentées s'agissent des images résultantes du pipeline JPEG2000.

Quantificateur		
Pas	Taille zone morte	Taux de compression
1	10	99.67744962704640%
4	10	99.69689952048823%
7	10	99.70292873518680%
4	4	99.69190460621911%
4	10	99.69689952048823%
4	16	99.70101149536633%
1	4	99.67073928767476%
7	16	99.70577936807776%

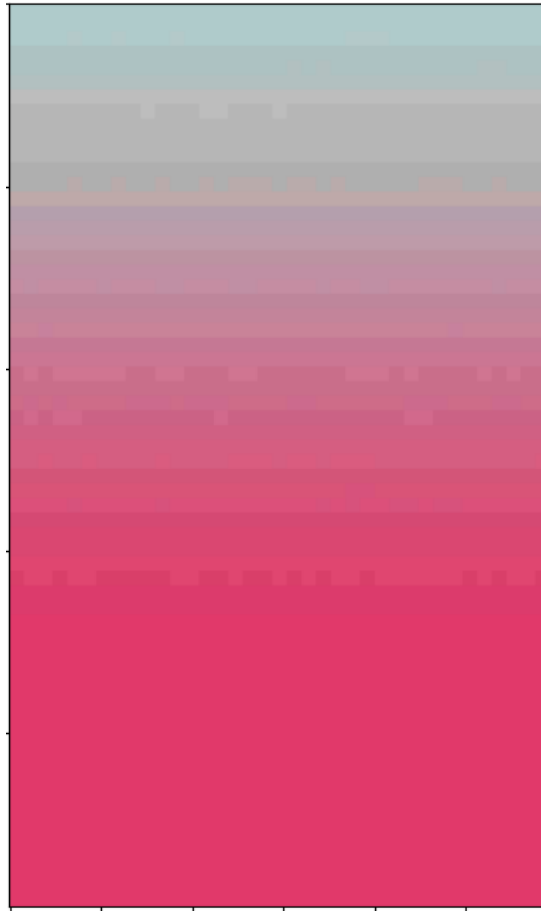
Voici le résultat avec **un pas de 1** et **une zone morte de taille 4** (meilleure qualité testée) :



Voici le résultat avec **un pas de 4** et **une zone morte de taille 10** (qualité moyenne) :



Voici le résultat avec **un pas de 7 et une zone morte de taille 16** (pire qualité testée) :



4.2)

On peut clairement constater des effets de bloc qui apparaissent, avec même des blocs qui dénaturent l'image originale. On avait un dégradé constant et lent, sans différences sur le plan horizontal. Après l'application du pipeline, on peut apercevoir de telles différences, soit des blocs d'une couleur différente sur une même zone horizontale. Bien entendu, ce n'est pas un exemple d'image qui met en valeur l'algorithme JPEG2000, mais cela nous permet justement d'en constater les limites. En théorie, on avait vu dans le cours que la conséquence normale de l'application de cet algorithme est une image plutôt floue. Cependant, dans notre cas, soit un cas quand même extrême, on retrouve carrément des effets de bloc qui apparaissent. Effectivement, plus on augmente le nombre de « **pas** » utiliser pour la quantification, plus la qualité visuelle de l'image résultante du pipeline se dégrade.

Avec des images plus régulières comme des photos, cette caractéristique est un peu plus camouflée, mais on peut quand même l'observer de façon assez évidente. On doit donc chercher à garder une valeur de « **pas** » assez bas pour conserver une bonne qualité, surtout considérant que le taux de compression est excellent dans tous nos tests effectués. La **taille de la zone morte**, en revanche, ne semble pas avoir un impact très important sur la qualité de l'image produite.

Question 5

Image	Taux de compression
Noir et blanc	99.38432713191587%
Colorée	99.18306505772748%
Photo standard	99.60807576359902%
Lent dégradé	99.69689952048823%

Le taux de compression est calculé selon la formule vu en cours soit :

$$1 - T_c/T_o$$

Le quantificateur utilisé pour ce numéro est composé d'un « **pas** » de 4 et une **zone morte** de taille 10. On constate clairement les effets de la compression dans les images résultante du pipeline. Le meilleur taux de compression obtenu était pour l'image du dégradé, et le pire est pour l'image colorée. Il semble que plus la plage des couleurs est petite et que l'image est homogène, plus la compression est élevée. La taille du dictionnaire construit par l'encodage LZW peut varier avec ce facteur, et cela explique, en partie, les différences observées dans les taux de compression. Normalement, le sous-échantillonnage et le gain qui lui est associé ne devraient pas être affectés de façon significative par la présence de couleurs. La quantification est appliquée de la même façon pour les composantes Y, U et V, donc ce n'est pas un facteur non plus.

Question 6

La transformée par ondelettes (DWT) ne cause pas d'effets de blocs, ce qui est utile lorsqu'on a une image avec des petites variations de couleurs. Par exemple, pour une photo présentant le ciel, les teintes de bleu sont très similaires, mais il y a quand même des légères différences. Le changement de fréquences est conservé dans la transformation, ce qui fait que les variations sont traitées sur des plus petits échantillons plutôt que des grands (d'où l'effet de bloc). On observe, en revanche, du flou ajouté à l'image.

Références

Babcock, A. (2019, mars 4). *Chroma Subsampling 4:4:4 vs 4:2:2 vs 4:2:0 - RTINGS.com*. (RTINGS.com) Consulté le octobre 10, 2020, sur <https://www.rtings.com/tv/learn/chroma-subsampling>

Flores-Mangas, F. (2018, janvier 8). *CSC320L11*. Récupéré sur CSC320W: Introduction to Visual Computing: <https://www.cs.toronto.edu/~mangas/teaching/320/slides/CSC320L11.pdf>