

# INF8770

## Technologies multimédias

### H2020 - Travail pratique #2

#### Pipeline JPEG

---

#### Remise du travail :

- Au plus tard, le 13 mars 2020, 17h00 sur Moodle - **aucun retard accepté**.

#### Documents à remettre :

- Votre code source ainsi qu'un rapport en **PDF** dans une archive (.zip/.7z/...). Un rapport sous un autre format se verra appliquer une pénalité.
- Vous devrez remettre votre rapport sur turnitin.com. Les instructions sont sur Moodle.

#### Autres directives :

- Il vous est interdit d'utiliser du code écrit par d'autres équipes.
  - Il vous est permis d'utiliser du code trouvé sur internet et des librairies externes, mais vous devez nous donner toutes les références.
  - Le code lui-même n'est pas évalué, mais il doit être remis au complet.
- 

Travail à réaliser : Dans ce TP, vous devez implanter toutes les étapes du pipeline de compression jpeg, ainsi que les étapes inverses. Vous devez réaliser des expériences afin de répondre aux questions. Dans votre rapport, expliquez brièvement la provenance de votre code et le travail réalisé pour l'adapter. Vous devez aussi décrire les expériences réalisées et vos conclusions.

#### Pipeline jpeg, étapes à réaliser :

Note : Il vous est conseillé de développer l'inverse de chaque étape directement à la suite du codage de celle-ci, afin de pouvoir afficher l'image. Vous pourrez ainsi valider visuellement le bon fonctionnement de votre code, et aussi isoler chaque étape et son inverse afin de déboguer en cas de problème.

##### Étape 1) Conversion RGB/ $Y'C_B C_R$

Cette étape consiste à convertir l'espace de couleur d'une image de RGB vers  $Y'C_B C_R$ . Vous devez utiliser un sous-échantillonnage de 4 : 2 : 0. Vous pouvez utiliser les équations suivantes pour les conversions, si vous écrivez le code vous-même :

$$Y' = 0.299R + 0.587G + 0.114B$$

$$C_B = 128 + 0.564(B - Y')$$

$$C_R = 128 + 0.713(R - Y')$$

$$R = Y' + 1.403(C_R - 128)$$

$$G = Y' - 0.714(C_R - 128) - 0.344(C_B - 128)$$

$$B = Y' + 1.773(C_B - 128)$$

### Étape 2) Le découpage en bloc de pixel

Dans cette partie, vous devez diviser vos images en blocs de 16 x 16. Il vous est donc conseillé d'utiliser des images avec des tailles qui sont des multiples de 16.

### Étape 3) Transformée en cosinus discrète (DCT)

La transformée en cosinus discrète transforme une matrice du domaine spatial vers le domaine fréquentiel. Cela permet d'identifier facilement les différentes fréquences et les quantifier. Dans cette étape, vous devez appliquer la DCT à chaque bloc obtenu précédemment, à l'aide de cette équation :

$$C(u, v) = \alpha(u) \cdot \alpha(v) \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cdot \cos\left(\frac{\pi(2y+1)v}{2N}\right)$$

où

$$\alpha(w) = \sqrt{\frac{1}{N}} \text{ si } w = 0$$

$$\alpha(w) = \sqrt{\frac{2}{N}} \text{ sinon}$$

Pour l'opération inverse, vous devez utiliser cette équation :

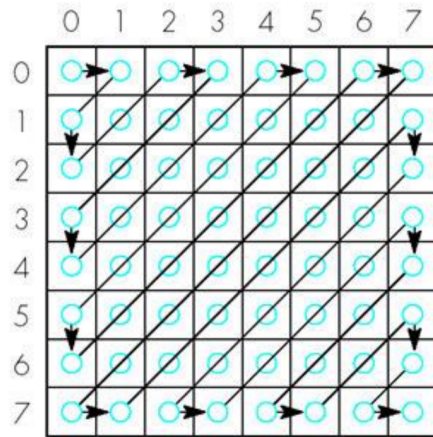
$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u) \cdot \alpha(v) \cdot C(u, v) \cdot \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cdot \cos\left(\frac{\pi(2y+1)v}{2N}\right)$$

### Étape 4) Quantification

Cette étape consiste à diviser terme par terme le bloc issue de la DCT par le terme correspondant dans la matrice de quantification. La matrice de quantification doit être de la même taille que le bloc DCT, donc 8x8 ici. Vous pouvez chercher sur internet pour trouver des matrices de quantification typiques, ainsi qu'expérimenter avec vos propres matrices. Notez que plus les valeurs dans cette matrice sont grandes, plus la qualité sera dégradée, et qu'une matrice composée de 1 ne dégrade donc pas du tout l'image.

### Étape 5) Zigzag

Vous devez ensuite parcourir les blocs quantifiés en zigzag pour créer des vecteurs 1D (comme dans la figure ci-dessous).



#### Étape 6) Encodage de Huffman et RLE

Pour finir la compression, les vecteurs 1D obtenus doivent être compressés avec l'encodage de Huffman et RLE.

#### Étape 7) Calcul du taux de compression

Ici, vous devez calculer et afficher le taux de compression de votre pipeline. Vous devez utiliser l'équation :

Taux = 1 - (Longueur du signal compressé / Longueur du signal original)

Attention, nous parlons ici d'une longueur en terme d'octets ou de bits, et non du nombre d'éléments dans un vecteur.

#### Étape 8) Le chemin inverse

Vous devez par la suite développer toutes les étapes inverses des méthodes que vous venez de coder afin de retrouver une image RGB et pouvoir l'afficher.

### Questions :

#### Question 1 (/10)

Discutez des effets positifs et négatifs du sous-échantillonnage 4 : 2 : 0 lors de la conversion RGB/Y'C<sub>B</sub>C<sub>R</sub>. Comparez avec au moins un autre sous-échantillonnage de votre choix. Les facteurs à discuter sont par exemples la qualité visuelle et le taux de compression. À votre avis, pourquoi fait-on un changement de l'espace de couleur avant de faire un sous-échantillonnage ?

#### Question 2 (/10)

Expliquez l'usage de la DCT dans ce contexte. La DCT seule permet-elle de compresser ?

#### Question 3 (/10, 5 par sous-question)

3.1) Discutez des artéfacts visibles ainsi que de leur forme, et de la cause de leur apparition.

3.2) Expérimentez avec diverses matrices de quantification, et présentez vos résultats (taux de compression) sous forme de tableau. Dans cette section, vous devez aussi inclure quelques exemples de vos images compressées.

**Question 4 (/10)**

Évaluez le taux de compression de votre algorithme sur différents types d'images (image noire, image très colorée, photo standard, lent dégradé). Présentez vos résultats sous forme de tableau. Discutez des raisons qui peuvent expliquer différents taux de compression.

**Question 5 (/10)**

Évaluez et comparez la perte de qualité accumulée sur plusieurs cycles de compression/décompression dans ces trois cas d'utilisation.

- Une matrice de quantification haute qualité (toujours la même utilisée).
- Une matrice de quantification basse qualité (toujours la même utilisée).
- Une alternance entre plusieurs matrices de quantification différentes.

**Question 6 (/10)**

Sur Moodle, récupérez le dossier *Fichiers\_TP2*. Il contient les images du jeu de données disponible à l'adresse suivante : [r0k.us/graphics/kodak/](http://r0k.us/graphics/kodak/). Comparez les performances de votre implémentation du pipeline JPEG sur ces images avec le pipeline JPEG2000. Vous n'avez pas à implémenter le pipeline JPEG2000, utilisez les images déjà compressées pour vous disponibles dans les dossiers *big* et *small*.

Évaluez notamment la taille et qualité des images compressées par les deux méthodes, grâce aux méthodes PSNR et SSIM, en fonction des matrices de quantification utilisées.