Logo

Description automatically generated

INF8770

Technologies multimédias

*Travail pratique #2 – Compression d’images avec la transformée de Karhunen-Loève*

Soumis par :

*Gaudet, Alexandre – 2090935*

*Sant’anna, Abdel – 2088865*

*Le 19 février 2023*

**Question 1**

L’objectif principal de l’application d’une transformée est de décorréler les données. Le principe général de la transformée KL est de concentrer un maximum de la variation des données selon un axe de grande énergie en appliquant un changement de base, tout en permettant de coder avec moins de bits l’axe de faible énergie, qui contient le moins de variation possible. Tout d’abord, il faut calculer la moyenne des triplets RGB (ou YUV) pour ensuite trouver la matrice de covariance propre à l’image. À partir de cette matrice, les vecteurs propres obtenus constituent la nouvelle base. Ceux-ci sont triés en ordre décroissant d’énergie (valeur propre). Finalement, le changement de base est appliqué sur chaque pixel de l’image. Une fois la changement de base effectué, la première composante contient la majorité de la variation des données alors que c’est l’inverse pour la troisième composante. Il est donc envisageable de réduire le nombre de bits utilisés pour les composantes de faible énergie afin d’atteindre un certain niveau de compression, tout en limitant la perte d’informations.

La différence principale entre la transformée KL et d’autres transformées comme DCT ou ondelettes est que avec KL, la matrice de covariance qui permet de calculer le changement de base est différente pour chaque image. Avec la transformée DCT, des cosinus à différentes fréquences sont utilisés. Pour la transformée en ondelettes, ce sont des ondelettes avec différentes fréquences et durées. Dans les deux cas, ces composantes sont connues à l’avance et ne dépendent pas de l’image. Ceci veut dire que la transformée KL introduit un coup de calcul supplémentaire et non négligeable.

**Question 2**

Après avoir appliqué l’algorithme proposé, dont le code est basé sur celui fourni dans le cadre du cours [1], sur les différentes combinaisons d’images, de niveaux de quantification et d’espaces colorimétriques, les mesures de qualité SSIM / PSNR ainsi que le taux de compression ont été mesurés. Les figures 1 et 2 présentent respectivement la mesure SSIM et PSNR obtenue à partir de l’image reconstruite en comparaison avec l’image initiale pour les 40 combinaisons . La figure 3 présente le taux de compression obtenu après quantification dans chaque cas.

Figure 1. Mesure de la qualité SSIM après reconstruction

Figure 2. Mesure de la qualité PSNR après reconstruction

Figure 3. Taux de compression après quantification

**Question 3**

En ce qui concerne l’espace colorimétrique, la figure 1 illustre clairement que pour la plupart des combinaisons d’images et de quantifications, RGB fournit une qualité SSIM équivalente ou supérieure à YUV après reconstruction de l’image. Le niveau de qualité PSNR indiqué à la figure 2 est plus partagé, RGB étant meilleur dans 55 % des cas. Cependant, en considérant seulement les niveaux de quantification qui offrent une certaine compression (8/8/4, 8/8/0, 8/4/4), RGB est le plus avantageux dans ~73 % des cas. L’espace de couleur RGB est donc celui qui offre généralement le meilleur niveau de qualité pour l’image reconstruite. En tenant aussi compte du fait que, contrairement à YUV, utiliser RGB n’introduit pas de calculs supplémentaires pour la conversion, il semble juste de conclure que l’espace de couleur RGB est celui qui offre les meilleurs résultats dans l’ensemble.

En ce qui concerne la quantification, identifier le meilleur choix parmi ceux considérés demande un peu plus d’analyse. Le premier niveau de quantification considéré, soit 8/8/8, n’offre pas de compression comparativement à une image RGB, aussi codée avec 8 bits par dimensions. Dans un contexte de compression d’image, ce choix ne semble pas optimal. À l’inverse, les niveaux de quantification 8/8/0 et 8/4/4 offrent tous les deux le meilleur taux de compression. Toutefois, en comparant les mesures de qualité pour ces deux quantifications avec RGB, 8/4/4 donne un meilleur PSNR dans tous les cas et un meilleur SSIM dans certains cas aussi (kodim05, kodim23). Pour mieux les départager, le tableau 1 présente la perte en PSNR et SSIM par pourcentage de compression obtenu, comparativement à la quantification 8/8/8 (aucune compression). La quantification 8/8/0 est la pire pour le PSNR dans tous les cas et la pire pour le SSIM dans trois cas. Ainsi, à taux de compression égal, 8/4/4 semble généralement offrir une meilleure qualité pour l’image reconstruite. Pour finir, il faut départager la quantification 8/8/4 et 8/4/4. En moyenne, la quantification 8/8/4 perd 0,31 PSNR et 0,0004 SSIM par pourcentage de compression obtenu, alors que pour 8/4/4 c’est une perte de 0,33 PSNR et 0,0006 SSIM. Purement en termes d’efficacité par pourcentage de compression, les deux niveaux de quantification sont donc quand même très similaires. À efficacité équivalente, le choix qui offre le plus de compression (8/4/4) semble plus intéressant, pourvu que les valeurs PSNR et SSIM demeurent acceptables. Comme le montrent les figures 1 et 2, la quantification 8/4/4 n’obtient jamais un SSIM inférieur à 0,945 et un PSNR inférieur à 37,08 dans l’espace RGB. Il s’agit là de valeurs quand même très bonnes. En surcroit, l’analyse visuelle des images reconstruites à partir de la quantification 8/4/4 ne permet pas d’identifier des artefacts visuels importants qui nuisent grandement à la qualité de l’image. Tout compte fait, le quantificateur 8/4/4 semble donc être le meilleur choix pour la compression des images.

En somme, l’analyse des résultats obtenus à la question 2 permet de conclure que la meilleure configuration est celle qui utilise l’espace colorimétrique RGB et la quantification 8/4/4. Ce sont donc ces paramètres qui seront considérés pour la suite.

Tableau 1. Perte PSNR et SSIM selon l'image et la quantification dans l'espace RGB

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | **Quantification** | **Perte PSNR par % de compression** | **Perte SSIM par % de compression** |
| kodim01 | 8/8/4 | 0.140 | 0.0000 |
| 8/8/0 | 0.415 | 0.0001 |
| 8/4/4 | 0.215 | 0.0001 |
| kodim02 | 8/8/4 | 0.066 | 0.0001 |
| 8/8/0 | 0.268 | 0.0001 |
| 8/4/4 | 0.242 | 0.0006 |
| kodim05 | 8/8/4 | 0.701 | 0.0007 |
| 8/8/0 | 0.745 | 0.0011 |
| 8/4/4 | 0.423 | 0.0006 |
| kodim13 | 8/8/4 | 0.144 | 0.0000 |
| 8/8/0 | 0.441 | 0.0001 |
| 8/4/4 | 0.351 | 0.0003 |
| kodim23 | 8/8/4 | 0.498 | 0.0011 |
| 8/8/0 | 0.826 | 0.0017 |
| 8/4/4 | 0.407 | 0.0016 |

**Question 4**

Suite à l’application de l’algorithme proposé, dont le code est basé sur celui fourni dans le cadre du cours [1], les mesures de qualité de l’image reconstruite et le taux de compression ont été pris en note pour chaque paire d’images *i1* et *i2*. Les tableaux 2 et 3 présentent respectivement la valeur SSIM et PSNR pour la reconstruction de l’image *i2*. Le tableau 4 présente le taux de compression obtenu.

Tableau 2. Mesure de la qualité SSIM en fonction de la paire d'images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i1 \ i2** | **kodim01** | **kodim02** | **kodim05** | **kodim13** | **kodim23** |
| **kodim01** | 0.995 | 0.981 | 0.980 | 0.988 | 0.947 |
| **kodim02** | 0.994 | 0.979 | 0.981 | 0.988 | 0.950 |
| **kodim05** | 0.995 | 0.982 | 0.978 | 0.988 | 0.951 |
| **Kodim13** | 0.996 | 0.982 | 0.980 | 0.988 | 0.939 |
| **Kodim23** | 0.995 | 0.980 | 0.981 | 0.988 | 0.945 |

Tableau 3. Mesure de la qualité SSIM en fonction de la paire d'images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i1 \ i2** | **kodim01** | **kodim02** | **kodim05** | **kodim13** | **kodim23** |
| **kodim01** | 45.03 | 43.25 | 37.93 | 39.40 | 37.84 |
| **kodim02** | 44.51 | 42.73 | 37.72 | 39.30 | 37.96 |
| **kodim05** | 44.87 | 43.55 | 37.06 | 39.18 | 37.85 |
| **Kodim13** | 45.52 | 43.77 | 37.27 | 39.28 | 37.03 |
| **Kodim23** | 44.86 | 43.14 | 37.98 | 39.11 | 37.58 |

Tableau 4. Taux de compression en fonction de la paire d'images

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i1 \ i2** | **kodim01** | **kodim02** | **kodim05** | **kodim13** | **kodim23** |
| **kodim01** | 0.333 | 0.333 | 0.333 | 0.333 | 0.333 |
| **kodim02** | 0.333 | 0.333 | 0.333 | 0.333 | 0.333 |
| **kodim05** | 0.333 | 0.333 | 0.333 | 0.333 | 0.333 |
| **Kodim13** | 0.333 | 0.333 | 0.333 | 0.333 | 0.333 |
| **Kodim23** | 0.333 | 0.333 | 0.333 | 0.333 | 0.333 |

**Question 5**

En observant les tableaux 2 et 3, il est possible d’identifier certaines tendances. Généralement, si une image *i1* offre la meilleure base pour la qualité de l’image *i2* reconstruite, la base obtenue avec l’image *i2* fait partie des meilleures pour l’image *i1*. L’inverse est aussi vrai dans le cas où *i1* soit la pire base pour *i2*. Cette observation est vraie tout autant pour le PSNR que pour le SSIM. Autre constat intéressant, le changement de base obtenu avec l’image *kodim01* semble assez polyvalent. En effet, pour une image *i2* donnée, ce changement de base n’est jamais le pire et fait souvent partie des meilleurs. Cette observation amène à une réflexion intéressante selon laquelle on pourrait considérer utiliser le changement de base de *kodim01* pour toutes les images, sous réserve que les données obtenues soient représentatives des images traitées par l’algorithme. Cela éliminerait le besoin de calculer pour chaque image le changement de base basé sur la matrice de covariance.

Références

[1] Bilodeau, G (2020) INF8770 [Code source]. <https://github.com/gabilodeau/INF8770/tree/master>