



**POLYTECHNIQUE
MONTRÉAL**

INF8770

Technologies Multimédias

Travail pratique #2

Compression d'images avec la transformée

Karhunen-Loève

Matthieu Basset — 2225981

Février 2024

Table des matières

| | | |
|----------|-------------------|----------|
| 1 | Question 1 | 2 |
| 2 | Question 2 | 3 |
| 3 | Question 3 | 4 |
| 4 | Question 4 | 5 |
| 5 | Question 5 | 6 |

1 Question 1

La transformée KL vise à passer une image dans la base des vecteurs propres de la matrice de covariance associée à la valeur de ses pixels après que ces derniers aient été centrés. En d'autres termes, cette opération permet de transformer les données afin de travailler sur les composantes contribuant le plus à la variance de l'image. Cette représentation est très utile, en effet travailler dans cet espace permet de plus facilement réduire la quantité de détails liée à des données apportant moins d'informations à l'image d'origine.

Pour effectuer cette transformation, on commence par centrer les données en leur soustrayant leur moyenne, cela permet d'éviter les biais lors de la recherche de vecteurs propres. On calcule ensuite la matrice de covariance de la liste de vecteurs obtenue. On peut ensuite obtenir ses vecteurs propres, ce qui permet de passer les données dans l'espace lié à ces vecteurs. On peut effectuer différentes opérations sur les données tant qu'elle sont dans cet espace. On peut ensuite retrouver l'espace de base en calculant la matrice pseudo inverse liée à la matrice de vecteurs propres et ajouter la moyenne soustraite auparavant. On retrouve ainsi l'image dans un espace similaire à celui d'origine.

Cette transformée se distingue des transformées DCT et en ondelettes par les éléments sur lesquels elle opère. Ces deux dernières décomposent le signal en somme de signaux periodiques, là où la transformée KL cherche les axes de plus grande variance dans l'image

2 Question 2

L'implémentation du TP est réalisée en python via un notebook. On trouve les résultats suivants pour la compression, utilisant des niveaux 8/8/8, 8/8/4, 8/8/0 et 8/6/4, et en utilisant l'espace colorimétrique RGB ou YUV:

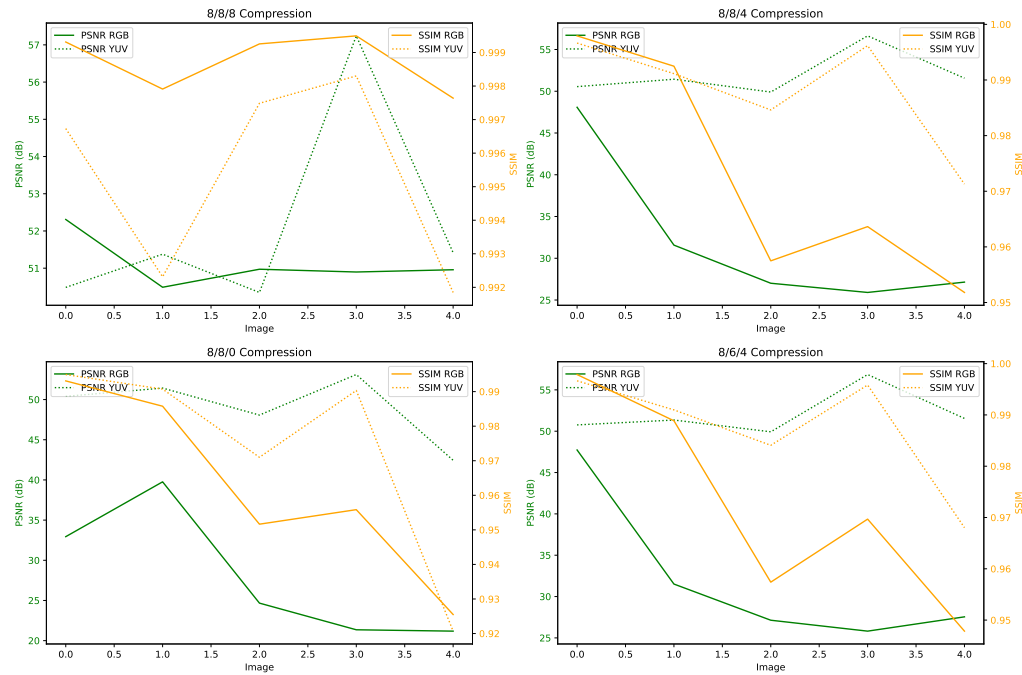


Figure 1: Résultats pour chaque image

3 Question 3

Lorsque l'on s'intéresse à la métrique de qualité couplé au taux de compression, ce sont les quantifications 8/8/0 qui offrent le compromis le plus satisfaisant. Certes la qualité de l'image est dégradée de manière visible, mais la compression offerte compense cette perte. On notera que par opposition aux autres méthodes, une quantification agressive sur une image dans l'espace de la transformée KL donnera des artefacts plus violents visuellement.

Pour ce qui est de l'espace de couleur le plus approprié, pour la métrique PSNR, l'espace YUV offre de meilleurs résultats, pour la métrique SSIM, les résultats sont plus proches. Ces métriques étant basées sur la perception humaine, il est naturel que l'espace YUV donne de meilleurs résultats. En effet cet espace est également conçu de manière à discriminer les composantes de couleur pour la compression adaptée à l'œil humain.

Ainsi, on peut conclure que la combinaison maximisant le rapport qualité/compression est une quantification 8/8/0 utilisée sur une image YUV.

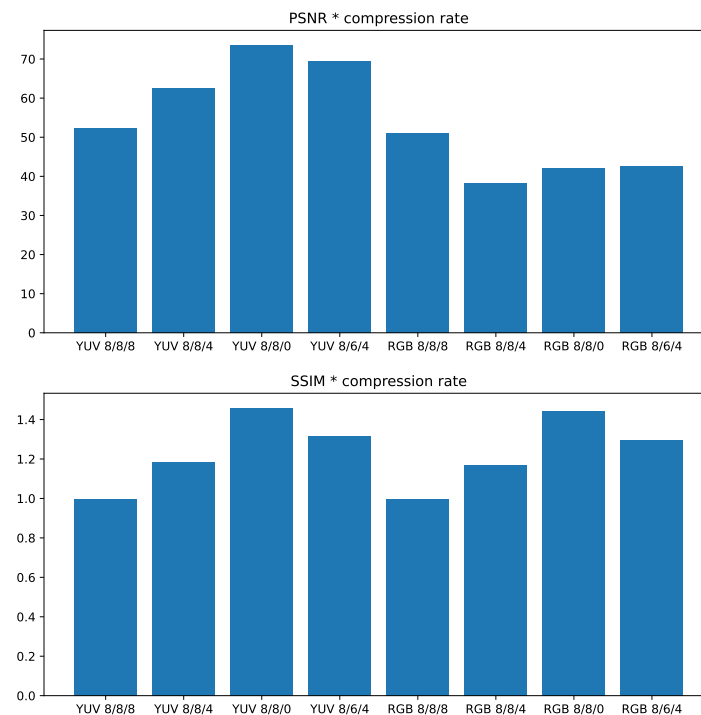


Figure 2: Résultats moyens qualité/compression pour chaque paramètre

4 Question 4

D'un point de vue implémentation, il suffit de découper l'algorithme précédent et de remplacer les matrices de vecteurs propre par celles d'autres images.

On obtient les résultats suivants pour la compression d'une image i en abscisse dans la base d'une image j en ordonnée.

Le taux de compression est de 33.3% pour chaque image car les niveaux de quantification 8/8/0 avaient été retenus à la question précédente.

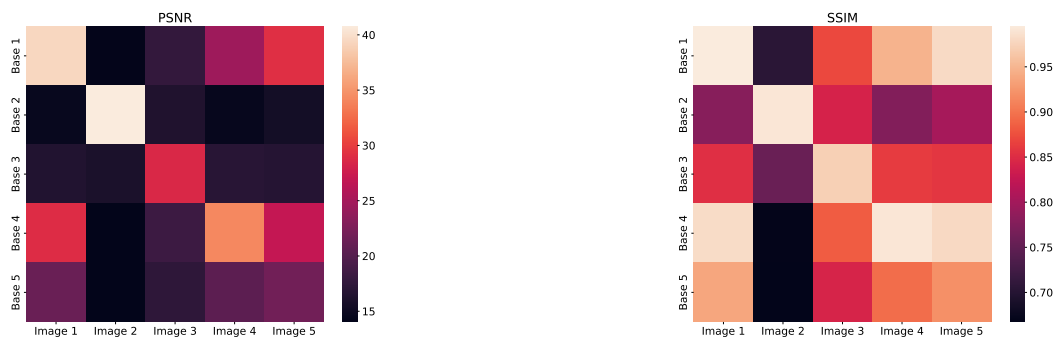


Figure 3: Valeurs de PSNR et SSIM pour les images quantifiées dans d'autres bases.

5 Question 5

On peut remarquer sans surprise que les résultats de qualité sont inférieurs sur les chaque image n'utilisant pas sa décomposition comme base pour la transformation. C'est d'ailleurs là une illustration d'un problème de la transformée KL, elle nécessite de calculer la base de vecteurs propre pour chaque image. Par opposition à DCT qui utilise une base standard de cosinus, on LZ77 que l'on peut utiliser sur des dictionnaires pré-construits.

On remarquera cependant que certaines combinaisons donnent des résultats plutôt satisfaisant, les images 1 et 4 utilisées comme base donnent des résultats acceptables pour les mesures de SSIM (excepté pour l'image 2 qui donne des résultats médiocres pour chaque autre base utilisée, sûrement dû à sa structure particulière très unie).