

Projet n°3

Compression d'image à travers la factorisation SVD

Groupe n°1 - Equipe n°2

Responsable : C. El-Habr

Secrétaire : C. Brouard

Codeurs : V. Ruello, A. Tirel

Résumé : Ce projet consiste à développer et optimiser des algorithmes permettant d'appliquer la factorisation SVD à une matrice. Pour ce faire on devra calculer la matrice de Householder permettant d'envoyer un vecteur sur un autre, opérer la bidiagonalisation d'une matrice, puis la factorisations QR de matrices bidiagonales. L'objectif final du projet est d'appliquer ces algorithmes à la compression d'images.

Transformations de Householder

Mise sous forme bidiagonale

Transformations QR

Application à la compression d'image

Le principe utilisé ici pour compresser une image est le suivant : on considère l'image à compresser sous la forme d'une matrice M de pixels (bitmap) de taille $w \times h$ avec w la largeur et h la hauteur de l'image. Chaque pixel est représenté par un triplet regroupant les trois composantes de couleurs rouge, vert et bleu.

Il s'agit ensuite de factoriser cette matrice sous la forme suivante :

$$M = U \times S \times V^*$$

avec :

- U une matrice de taille $w \times h$ unitaire (telle que $UU^* = U^*U = Id$) ;
- V une matrice de taille $h \times w$ unitaire ;
- S une matrice de taille $h \times h$, carrée et diagonale, dont les éléments diagonaux sont rangés par ordre décroissant et sont appelés valeurs singulières de M .

Compresser l'image au rang k consiste alors à annuler tous les éléments diagonaux de S d'indice supérieur ou égal à k .

Voir sur la figure 1 un exemple d'application de notre programme : une comparaison de l'image de fusée fournie et sa version compressée au rang 10.

FIGURE 1 – Image originale à gauche, image compressée avec $k = 10$ à droite.

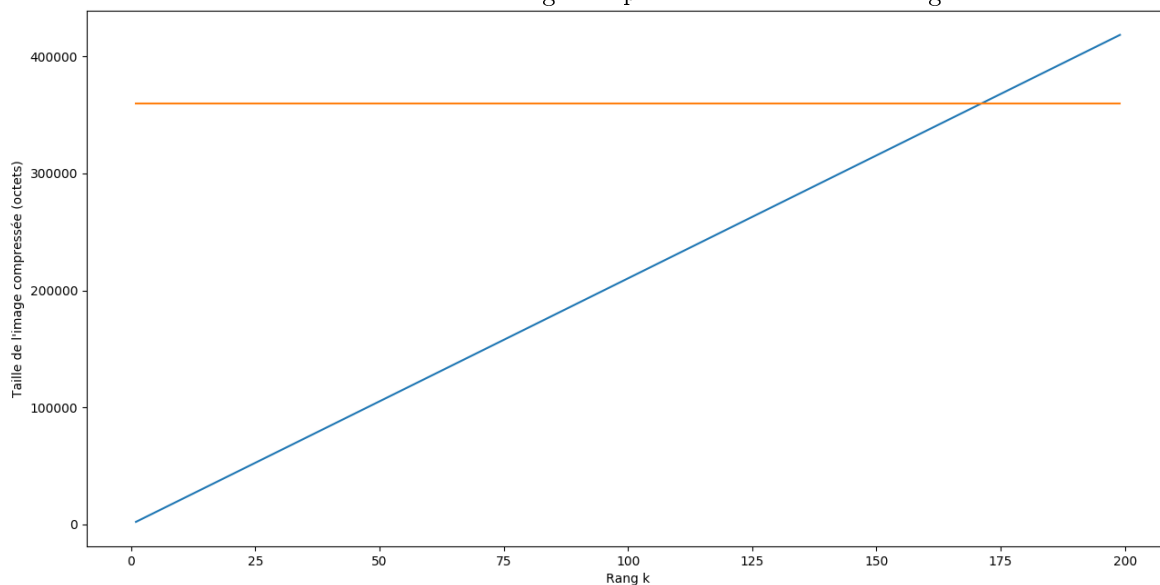


La taille du fichier une fois l'image compressée au rang k , est proportionnelle à k , ce qui implique évidemment que la taille de l'image compressée peut dépasser celle de l'image d'origine à partir d'un certain rang.

Le graphe présenté sur la figure 2 montre l'évolution de la taille de l'image compressée en fonction du rang (courbe bleue) et la taille de l'image d'origine (ligne orange), qui est la même image de fusée utilisée précédemment.

On observe que la taille de l'image compressée dépasse celle de l'image d'origine pour des rangs supérieurs à 170 environ.

FIGURE 2 – Taille de l'image compressée en fonction du rang.



Si on note k_{max} le rang tel qu'une compression au rang k_{max} produit un fichier de taille égale à celle de l'image d'origine, on obtient l'efficacité e suivante :

$$e = \frac{k_{max} - k}{k_{max}}$$

Les figures 3 et 4 montrent qu'il est possible de garder une qualité d'image tout-à-fait satisfaisante même en compressant à un rang arbitrairement bas, donc avec une efficacité intéressante. En effet, on peut voir que les valeurs singulières et les erreurs algébriques pour les trois composantes de couleurs, convergent très vite vers 0.

FIGURE 3 – Valeur des 50 premières valeurs singulières des matrices r , g et b .

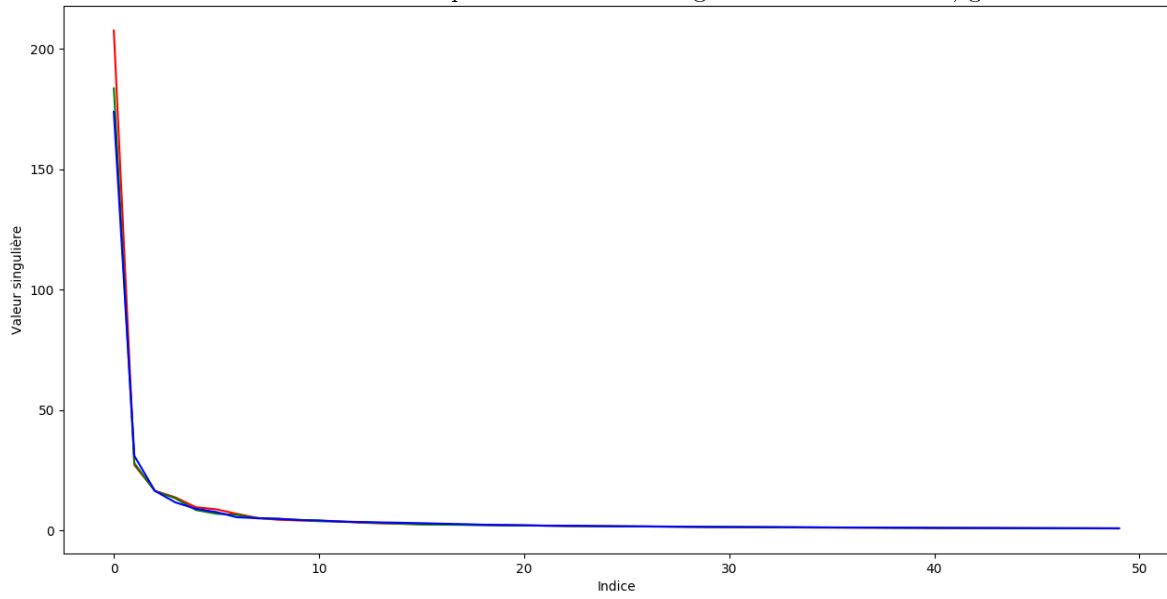


FIGURE 4 – Erreur algébrique selon les 3 composantes en fonction du rang.

