Soutenance de projet

Par Katell Nio, Charlotte Prouzet, Elsa Catteau et Safia Zaari Jabri

Compression d'images par transformée de Fourier

Polytech Nice Sophia

Objectif :

A partir d’une image donnée, créer un programme python qui compresse l’image puis la décompresse

Sommaire

1. Introduction
2. Transformée de cosinus discrète (DCT)
3. Compression et décompression
4. Comparaison erreur et taux de compression en fonction de la fréquence
5. Comparaison des images en fonction de la fréquence
6. Comparaison des images avec et sans quantification
7. Conclusion

Introduction

3 composantes de couleurs: RGB (red, green, blue)

Transformée de cosinus discrète (DCT)

DCT d'une image M de taille 8x8 :

$$D_{k,l} = \frac{1}{4} C_k C_l \sum_{i=0}^{7} \sum_{j=0}^{7} M_{i,j} \cos\left(\frac{(2i+1)k\pi}{16}\right) \cos\left(\frac{(2j+1)l\pi}{16}\right)$$

$$D = PMP^T$$

Détermination de la matrice P :

$$P = (p_{k,l})_{0 \leq k,l \leq 7} = \left(\frac{1}{2} C_k \cos\left(\frac{(2l+1)k\pi}{16}\right)\right)_{0 \leq k,l \leq 7}$$

$$Q = (q_{k,l})_{0 \leq k,l \leq 7} = \left(\frac{1}{2} C_l \cos\left(\frac{(2k+1)l\pi}{16}\right)\right)_{0 \leq k,l \leq 7} = P^T$$

Image avec une seule couleur

Composante bleue

Composante verte

Composante rouge

Compression

- M = matrice représentant un bloc 8 x 8 de l'image
- Division de la matrice $D = P\,M\,P^T$ par la matrice de quantification Q
- Garder uniquement la partie entière

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 13 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Décompression

- Multiplication par Q de chacun des blocs 8x8 de l'image compressée
- Appliquer le changement de base inverse : $\tilde{M} = P^T \tilde{D} \ P$
- Rassembler les blocs pour obtenir l'image originale

Image décompressée de pns_original avec F=6

Filtrage des hautes fréquences et débruitage

- On met à 0 tous les coefficients de la matrice D dont les indices (i, j) vérifient :

$$i + j \geqslant F$$

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{pmatrix}$$

Matrice 8 x 8 répertoriant la somme des indices i et j

> filtrage en complément de le la quantification ⟶ CodeCourt
> ou filtrage seul sans quantification ⟶ CodeCourt_filtrage_seul

Comparaison erreur et taux de compression en fonction de la fréquence pour pns_original.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	96.06	15
F = 4	88.51	9
F = 6	78.38	7
F = 15	44.42	25

Comparaison erreur et taux de compression en fonction de la fréquence pour pns_noisy.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	97.10	20
F = 4	91.64	16
F = 6	84.39	15
F = 15	58.95	20

Comparaison erreur et taux de compression en fonction de la fréquence pour pns_noisy.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	97.1[illegible]	20[illegible]
F = 4	91.64	16
F = 6	[illegible]	[illegible]
F = 15	58.95	20

Comparaison erreur et taux de compression en fonction de la fréquence pour trefles.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	95.5	31
F = 4	85.7	22
F = 6	72.9	16.5
F = 15	38.24	34

Comparaison erreur et taux de compression en fonction de la fréquence pour trefles.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	95.5	31
F = 4	[illegible]	22
F = 6	[illegible] 9	[illegible]
F = 15	38.[illegible]	34

Comparaison erreur et taux de compression en fonction de la fréquence pour vagues.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	96.32	1.5
F = 4	91	2.3
F = 6	82.37	4.9
F = 15	48.79	27

Comparaison erreur et taux de compression en fonction de la fréquence pour vagues.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	[illegible]	1.5
F = 4	91[illegible]	[illegible]3
F = 6	[illegible]	[illegible]9
F = 15	48.79	2[illegible]

Comparaison entre le code avec quantification et filtrage et le code avec filtrage seul

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	97.10	20
F = 4	91.64	16
F = 6	84.39	15
F = 15	58.95	20

Tableau répertoriant les valeurs trouvées avec le code “complet” (quantification + filtrage des hautes fréquences
Image : pns_noisy.png

Fréquence de coupure	Taux de compression (en %)	Erreur (en %)
F = 2	96.62	19.5
F = 4	88.96	15.8
F = 6	76.53	14.4
F = 15	26.98	0.4

Tableau répertoriant les valeurs trouvées avec le code du filtrage des hautes fréquences seul
Image : pns_noisy.png

Comparaison des images en fonction de la fréquence

Zoom de l’image original

Zoom de l’image avec F=6

Zoom de l’image avec F=3

Comparaison des images avec et sans quantification

Pour l'image vagues.png avec F = 6 :

Image avec filtrage seulement

Image avec quantification + filtrage

Comparaison des images avec et sans quantification

Pour l’image trefles.png avec F = 6 :

Image avec filtrage seulement

Image avec quantification + filtrage

Comparaison des images avec et sans quantification

Pour l'image pns_noisy.png avec F = 6 :

Image avec filtrage seulement

Image avec quantification + filtrage

Conclusion

- Filtrage des hautes fréquences : élimine le bruit en réduisant la taille des données tout en gardant les informations essentielles.

  → Meilleure qualité d'image

- Quantification : réduit davantage la taille en approximant les coefficients mais peut augmenter l'erreur relative.

  → Meilleure compression

- Faire varier la fréquence pour avoir un bon compromis entre qualité de l'image et compression.

MERCI POUR VOTRE ATTENTION !