



Facult  des sciences et techniques, Beni Mellal

Computer Vision - TP3

R alis  par :

- Bel-assal Mohamed

Encadr  par :

- Pr. Elayachi Rachid

Ann e acad mique : 2023-2024

1 EQM et PSNR

1.1 Erreur quadratique moyenne

L'Erreur Quadratique Moyenne (EQM), ou Mean Squared Error (MSE) en anglais, est une mesure de la qualité d'une image après un traitement (compression, filtrage, etc.). Elle quantifie la différence entre une image originale $I(x, y)$ et une image approximée $\hat{I}(x, y)$. Plus l'EQM est faible, plus l'image modifiée est proche de l'originale.

La formule de l'EQM est la suivante :

$$EQM = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N \left[I(x, y) - \hat{I}(x, y) \right]^2$$

où :

- M et N sont les dimensions de l'image (nombre de lignes et de colonnes),
- $I(x, y)$ est la valeur du pixel à la position (x, y) dans l'image originale,
- $\hat{I}(x, y)$ est la valeur du pixel à la position (x, y) dans l'image approximée.

Une faible valeur de l'EQM indique que l'image approximée est proche de l'image originale, tandis qu'une valeur élevée indique une grande différence, donc une perte de qualité importante.

```
def EQM(image_originale, image_reconstruite):  
  
    if image_originale.shape != image_reconstruite.shape:  
        raise ValueError("Taille diff")  
    diff = image_originale - image_reconstruite  
    eqm = np.mean(np.square(diff))  
  
    return eqm
```

1.2 Rapport signal-bruit de crête

Le PSNR (Peak Signal-to-Noise Ratio) est une mesure utilisée pour évaluer la qualité d'une image en comparant l'image dégradée avec l'image originale. Il est exprimé en décibels (dB) et calculé à partir de l'Erreur Quadratique Moyenne (EQM).

La formule du PSNR est donnée par :

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{EQM}} \right)$$

où MAX_I est la valeur maximale d'un pixel et EQM est l'Erreur Quadratique Moyenne entre les deux images.

Un PSNR élevé indique une faible différence entre les images, signifiant une meilleure qualité. Un PSNR faible indique une grande différence, donc une qualité dégradée.

```
def PSNR(image1, image2, max_pixel=255.0):  
    eqm = EQM(image1, image2)  
    if eqm == 0:  
        return float('inf')  
  
    psnr = 20 * np.log10(max_pixel / np.sqrt(mse))  
    return psnr
```

2 Transformation de Haar

La transformation de Haar est une technique de compression d'images basée sur les ondelettes. Elle décompose une image en coefficients représentant différentes fréquences. L'image est d'abord divisée en sous-blocs, puis chaque sous-bloc est transformé en utilisant la matrice de Haar. Les coefficients obtenus sont ensuite quantifiés pour réduire la taille des données, et enfin, les coefficients quantifiés sont codés de manière efficace pour la compression.

La matrice de Haar pour un bloc 2×2 est donnée par :

$$H = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

Cette matrice est appliquée à chaque sous-bloc de l'image pour obtenir les coefficients de transformation.

On utilise :

$$Sc_n = (A_h)^T \cdot M_{h-1} \cdot A_h$$

où h est le niveau de la compression, A_h est la matrice des coefficients de transformation (Matrice de Haar), M_{h-1} est l'image au niveau $h - 1$, et Sc_n est l'image comprimée.

Les coefficients A_{ij} de la matrice de Haar sont définis comme suit :

$$\begin{aligned} A_{ij} &= \frac{1}{2^n} && \text{si } j \leq \frac{N}{2^n} \text{ et } i = 2^n \cdot j + 1 \text{ ou } i = 2^n \cdot j \\ A_{ij} &= \frac{1}{2^n} && \text{si } \frac{N}{2^n} < j \leq N \text{ et } i = 2^n \cdot \left(j - \frac{N}{2^n} \right) - 1 \\ A_{ij} &= -\frac{1}{2^n} && \text{si } \frac{N}{2^n} < j \leq N \text{ et } i = 2^n \cdot \left(j - \frac{N}{2^n} \right) \\ A_{ij} &= 1 && \text{si } i > N \\ A_{ij} &= 0 && \text{sinon} \end{aligned}$$

3 Implementation

Nous avons une image que nous voulons compresser, on la convertit d'abord en niveaux de gris. voici l'image :



FIGURE 1 – Image utilisée

Pour commencer, nous allons mettre en œuvre la fonction qui nous permettra d'obtenir les coefficients de Haar. Cette fonction est essentielle pour transformer notre image en une représentation plus compacte.

```
def coef_haar(A):
    N, M = A.shape
    for i in range(N):
        for j in range(M):
            if j <= N / 2 and i == 2 * j + 1 or i == 2 * j:
                A[i, j] = 1 / 2
            elif j > N / 2 and j <= N and i == 2 * (j - N / 2) - 1:
                A[i, j] = 1 / 2
            elif j > N / 2 and j <= N and i == 2 * (j - N / 2):
                A[i, j] = -1 / 2
            elif i > N:
                A[i, j] = 1
            else:
                A[i, j] = 0
    return A
```

Ensuite, nous allons implémenter la fonction qui utilisera ces coefficients pour compresser l'image.

```
def compression_haar(IM, A, niveau):
    SC = IM.copy()
    i = 1
    for k in range(niveau):
        SC = np.dot(A.T, np.dot(SC, A))
    return SC
```

Dans ce cas, nous l'utiliserons en trois niveaux.



FIGURE 2 – Les 3 niveaux de compression

Maintenant, nous allons essayer de décompresser l'image. La décompression nous permettra de reconstruire l'image à partir des coefficients de Haar que nous avons obtenus précédemment.

```
def decompression_haar(IM, A, niveau):
    ID = IM.copy()
    invA = np.linalg.pinv(A)
    invAT = np.linalg.pinv(A.T)
    for k in range(niveau):
        ID = np.dot(invAT, np.dot(ID, invA))
    return ID
```

Voici le resultat obtenu :

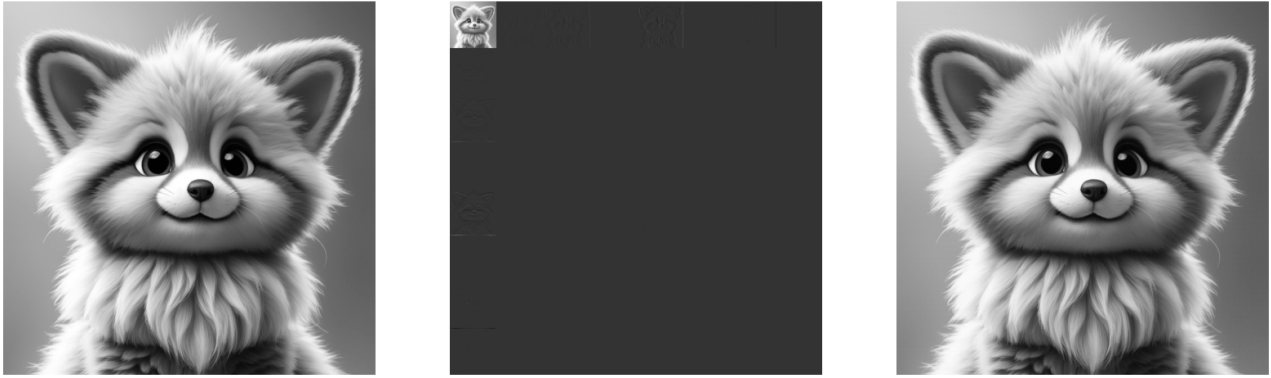


FIGURE 3 – De l'origine a la decompression

4 Mesure de performance

Nous utilisons les fonctions déjà définies, EQM (Erreur Quadratique Moyenne) et PSNR (Peak Signal-to-Noise Ratio), afin d'évaluer les performances des processus de compression et de décompression. Ces métriques sont essentielles pour mesurer la qualité des données après compression. L'EQM quantifie l'écart moyen au carré entre les valeurs originales et les valeurs reconstruites, tandis que le PSNR exprime la qualité de la reconstruction en termes de rapport logarithmique entre la puissance maximale du signal et l'erreur. Ces deux indicateurs permettent ainsi de juger la fidélité et l'efficacité de la méthode utilisée.

Qualité	Très bonne	Bonne	Acceptable	Mauvaise
PSNR (dB)	≥ 40	40-35	35-30	< 30

TABLE 1 – Tableau des qualités d'image selon les valeurs de PSNR.

Nous utilisons ce tableau 1 comme référence pour les valeurs de PSNR. Il nous permet d'évaluer la qualité d'image obtenue après compression et décompression en fonction des différentes plages

```
print("EQM: ", EQM(G_image, decompressed_image))
print("PSNR:", PSNR(G_image, decompressed_image))
```

L'EQM a donné la valeur : 1.7307, et le PSNR a donné la valeur : 45.7485. Étant donné que le PSNR est supérieur à 40, nous pouvons conclure que la compression à 3 niveaux ici est de très bonne qualité.