

## TP. Compression d'image

Le but de ce TP est d'implémenter une chaîne complète de compression/décompression d'image. Tous les deux cas – compression sans pertes et avec pertes – seront considérés. A fin d'être compressée, l'image est décorrélée, quantifiée (seulement dans la compression avec pertes) et, finalement, codée par un codage de source. La décompression est la succession inverse de ces opérations.

Pour decorrélation, on utilisera la prédiction linéaire dans la compression sans pertes et la DCT dans compression avec pertes. La chaîne avec pertes va inclure entre la DCT et le codage, une étape de quantification uniforme. Pour le codage, on utilisera, dans tous les deux cas, le codage de Huffman.

On observera l'influence de l'image sur le taux de compression. Pour la compression avec pertes, on étudiera la dépendance entre le taux de compression, le pas de quantification et la qualité d'image (PSNR) décompressée. Une comparaison entre les deux chaînes sera faite.

### COMPRESSION SANS PERTES

Dans la compression sans pertes, on ne fait pas de la quantification. La compression s'obtient à la fin de decorrélation et codage, qui sont toutes les deux des opérations réversibles.

**Images à traiter :** *Lena* et *Mandrill* de gamme dynamique de [0 31] ; ces deux images ont des entropies différentes. Pour afficher de manière visible ces images, il faut utiliser soit *imagesc* soit *imshow(Lena, [0, 31])*.

### COMPRESSION

1. **Decorrélation par prédiction:** en utilisant le prédicteur ci-dessous, remplacez chaque pixel de l'image par l'erreur de prédiction. La première ligne et la première colonne de l'image doivent être conservées à fin de pouvoir restaurer l'image par transformation inverse.

Ecrivez un program  $I\_PRED = pred(I, a_A, a_B, a_C)$  où  $I$  est l'image à compresser et  $I\_PRED$  est l'image décorrélée. Affichez l'image  $I\_PRED$  et dites en quoi son aspect est différent par rapport à l'image d'origine. Affichez l'histogramme de l'image avant et après la decorrélation et commentez les modifications.

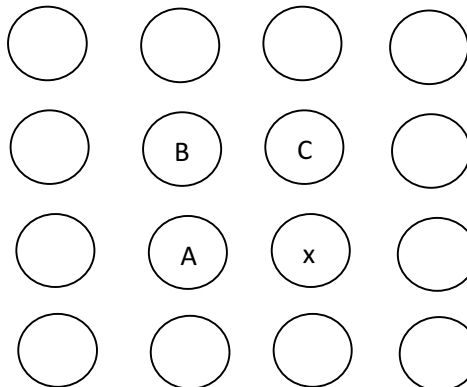
$$x_{ERR} = x - \hat{x}$$

$$\hat{x} = a_A A + a_B B + a_C C$$

$$a_A = 1$$

$$a_B = -1$$

$$a_C = 1$$



2. **Mesurer l'effet de la prédiction sur la corrélation:** Afin de mesurer l'effet de prédiction au niveau de la corrélation, calculez le coefficient de corrélation sur l'horizontale et verticale avant et après le calcul de l'erreur de prédiction et tracez la courbe du coefficient vs. écart pour des écarts de 0 à 10 (l'écart est la distance entre les colonnes/lignes prises en compte dans le calcul du coefficient de corrélation). Fonction Matlab à utiliser: *corrcoef*. Commentez les différences.
3. **Codage de Huffman:** écrivez un program `[I_C, dictionnaire]=codage(I_PRED)` qui, par codage de Huffman, fait la conversion de I\_PRED dans un vecteur binaire I\_C. Fonctions Matlab à utiliser pour création du dictionnaire et le codage: *huffmandict* et *huffmanenco*. Quel est la longueur moyenne des mots de code ? Calculez le taux de compression en utilisant la longueur moyenne des mots de code. Calculez l'entropie de l'image décorrelée et comparez-la avec la longueur moyenne des mots de code. Expliquez le résultat.

## DECOMPRESSION

4. **Décodage de Huffman:** écrivez un program `I_PRED=décodage(I_C, dictionnaire, lignes, colonnes)` qui, à l'aide du dictionnaire, transforme le vecteur binaire I\_C dans l'image I\_PRED. Il faut connaître la dimension de l'image pour reformater le vecteur décodé dans une image.
5. **Restauration de l'image:** écrivez un program `I_R=INVpred(I_PRED,A,B,C)` qui refait l'image à partir de l'image I\_PRED. Affichez l'image refaite. A fin de vérifier que la reconstruction est parfaite, calculez et affichez la différence  $DIFF=I-I_R$ .

## COMPRESSION AVEC PERTES

1. **Decorrelation par DCT :** Ecrivez un programme `I_DCT=DCTransf(I)` ou I est l'image à compresser et I\_DCT est l'image décorrelée. Traitez l'image sur des blocs adjacents de 8x8, comme dans le cours. Chaque bloc doit être transformé en utilisant la DCT 2D.
2. **Quantification :** Dans le bloc de 8x8 transformé, chaque coefficient est quantifié avec un pas de quantification qui dépend de sa position dans le bloc. Les pas de quantification sont indiqués dans le tableau ci-dessous. L'opération de quantification consiste dans le calcul suivant :

$$I_{Q(i,j)} = \text{round} \left( \frac{I_{DCT(i,j)}}{\text{pasDeQuantification}} \right)$$

Ecrivez un programme `I_Q=Quantif(I_DCT)` ou I\_Q est l'image quantifiée.

Pas de quantification dans un bloc de 8x8.

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

3. **Codage de Huffman**: de même que dans la compression sans pertes. Calculez le taux de compression et l'entropie de l'image décorrelée.
4. **Décodage de Huffman** : de même que dans la compression sans pertes.
5. **Restauration de l'image** : de même que dans la compression sans pertes mais en utilisant la DCT inverse.
6. **Qualité de la compression** Affichez l'image restaurée (commentez les distorsions) et évaluez sa qualité en calculant le PSNR de l'image restaurée (en utilisant l'expression du PSNR dans le cours).

**Comparez la compression avec perte avec la compression sans pertes** en termes de qualité de l'image restaurée et de taux de compression.

Téléchargez à

[https://drive.google.com/drive/folders/1NxGCFerbKuetBrgzjZrxzd\\_rvs4Z51CM?usp=sharing](https://drive.google.com/drive/folders/1NxGCFerbKuetBrgzjZrxzd_rvs4Z51CM?usp=sharing)

le compte-rendu dans un seul fichier pdf (les programmes Matlab inclus). Date limite **11 Decembre**.